

**UNCLASSIFIED**



**TNO-report**

**TNO-DV1 2004 A234**

**Model selection and accounting for uncertainty**

Oude Waalsdorperweg 63  
PO Box 96864  
2509 JG The Hague  
The Netherlands

www.tno.nl

Phone +31 070 374 00 00  
Fax +31 070 328 09 61  
Info-DenV@tno.nl

Date	May 2005
Authors	Ir. C.V. van Wijk Dr. habil. H.W.L. Naus
Classification	Unclassified
Affiliation	-
Project officer	Maj L. Lagerwerf
Classification date	-
Title	Unclassified
Management summary	Unclassified
Abstract	Unclassified
Report text	Unclassified
Appendices	Unclassified
Contract number	-
Sponsor	Ministry of Defence
Affiliation	-
Project name	Invloed van EM golven landmijnen
Project number	015.33826
Copy no	15
No of copies	24
No of pages	148
No of appendices	-

**DISTRIBUTION STATEMENT A**  
Approved for Public Release  
Distribution Unlimited

All information which is classified according to Dutch regulations shall be treated by the recipient in the same way as classified information of corresponding value in his own country. No part of this information will be disclosed to any third party.

The CLASSIFICATION designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret. All rights reserved. No part of this report may be reproduced in any form by print, photoprint, microfilm or any other means without the previous written permission from TNO.

In case this report was drafted on instructions from the Ministry of Defence the rights and obligations of the principal and TNO are subject to the standard conditions for research and development instructions, established by the Ministry of Defence and TNO, if these conditions are declared applicable, or the relevant agreement concluded between the contracting parties.

**20060808086**

© 2005 TNO

Netherlands Organisation for Applied  
Scientific Research (TNO)

AQ F06-11-7960

**UNCLASSIFIED**

# Modelselectie en het rekening houden met onzekerheden

## Introductie

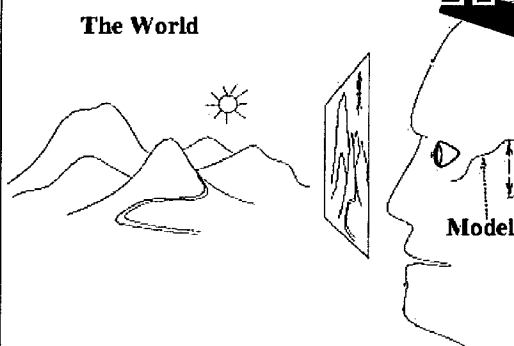
In een voorgaande studie heeft TNO Defensie en Veiligheid, locatie Den Haag in opdracht van het Ministerie van Defensie algoritmen ontwikkeld die doelen met een specifieke magnetische signatuur kunnen detecteren. Deze algoritmen bepalen een aantal parameters die karakteristiek zijn voor de betreffende signatuur. Bij praktische implementatie van de algoritmen dient zich een belangrijke vraag aan: 'Met hoeveel parameters moeten we rekening houden?' Dit zogenaamde modelselectieprobleem, een essentieel onderdeel van de algoritmen, is in de voorgaande studie niet aan de orde gesteld.

Modelselectie onderscheidt zich als één van de belangrijkste problemen binnen het vakgebied van de statistische deductie. Het is een activiteit met de bedoeling regels en beperkingen te leren aan de hand van gemeten data. Uiteindelijk wordt een hypothese geaccepteerd dan wel verworpen. Een voorbeeld van een dergelijke hypothese is: 'het aantal parameters voor voertuig A is 10.'

## Modelselectie

Het statistisch modelleren houdt zich bezig met het vinden van algemene regels uit waargenomen data. In het kort komt dit neer op het extraheren van informatie uit beschikbare data. In het modelleren moeten we niet meer toelaten dan strikt noodzakelijk, dat wil zeggen 'hebben we de keuze tussen indifferente alternatieven, dan moeten we de simpelste kiezen.'

Het doel van modelselectie is het zoeken van regelmatigheden in de data. 'Regelmaat' kan worden geïdentificeerd met 'de mogelijkheid tot compressie.' Het combineren van deze twee begrippen moet ons in staat stellen om, gegeven een verzameling van hypothesen en data, de hypothese te vinden die de data het meest



comprimeert. Hiertoe beschrijven we een principe genaamd 'minimum description length', dat gebaseerd is op het zoeken van regelmaat in data. Dit principe sluit een compromis tussen aanpassingsgraad van de data en de complexiteit van het model.

## Betrouwbare deducties

Met betrouwbare deducties kunnen we goede voorspellingen en beslissingen nemen met betrekking tot de data. Het stelt ons in staat om te bepalen wanneer we simpelere modellen kunnen gebruiken en wanneer niet. In het algemeen zijn we dus geïnteresseerd in wat al dan niet betrouwbaar kan worden voorspeld met een deels correct model.

In het rapport beschrijven we een nieuwe procedure genaamd 'entropificatie'. We kunnen met een 'geëntropificeerd' model, mits we over genoeg data beschikken, een model vinden met de kleinste fout. Tevens geeft het een juiste schatting van de gemiddelde fout. Het geeft dus een goede indruk 'hoe goed het werkelijk is'.

## Toepassing

Met de beschreven technieken is het mogelijk om voertuigsignaturen met grote zekerheid te herkennen uit gemeten data. Het uitvoeren van een groot aantal signatuurmetingen leidt tot a-priori kennis,



de Estimator

TNO-DV1 2004 A234

Mei 2005

Ir. C.V. van Wijk

Dr. habil. H.W.L. Naus

Ongerubriceerd

# Modelselectie en het rekening houden met onzekerheden

waardoor deductie nog betrouwbaarder wordt.

Tenslotte wordt opgemerkt dat de theorie algemeen toepasbaar is en niet alleen op magnetische voertuigsignaturen.

## PROGRAMMA

## PROJECT

Programmabegeleider  
Maj L. Lagerwerf,  
OTC Genie/Kenniscentrum

Projectbegeleider  
Kap C.M. Netten, OTC  
Genie/Kenniscentrum

Programmaleider  
Dr. A.J. Schoolderman,  
TNO Defensie en Veiligheid

Projectleider  
Dr.ir. S.H.J.A. Vossen,  
TNO Defensie en Veiligheid

Programmatitel  
Mijnen

Projecttitel  
Invloed van EM-golven op landmijnen

Programmanummer  
V010

Projectnummer  
015.33826

Programmaplanning  
Start 01-01-2001  
Gereed 31-12-2004

Projectplanning  
Start 01-01-2003  
Gereed 31-12-2004

Frequentie van overleg

Projectteam  
Dr.habil. H.W.L. Naus  
Dr.ir. S.H.J.A. Vossen  
Ir. C.V. van Wijk

ONGERUBRICEERD

## TNO Defensie en Veiligheid

Lokatie 's-Gravenhage

Oude Waalsdorperweg 63  
2597 AK 's-Gravenhage  
Postbus 96864  
2509 JG 's-Gravenhage

www.tno.nl  
Info-DenV@tno.nl

T 070 374 00 00  
F 070 328 09 61

ONGERUBRICEERD

## Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Kolmogorov complexity and information theory</b>	<b>9</b>
2.1	Overview and summary . . . . .	10
2.2	The coding framework and the Kraft inequality . . . . .	11
2.3	Shannon entropy versus Kolmogorov complexity . . . . .	14
2.3.1	Shannon entropy . . . . .	14
2.3.2	Kolmogorov complexity . . . . .	19
2.3.2.1	Formal details . . . . .	20
2.4	Universal coding: interpolating between Kolmogorov and Shannon . .	23
2.5	Mutual information . . . . .	27
2.5.1	Shannon mutual information . . . . .	27
2.5.2	Algorithmic mutual information . . . . .	30
2.6	Shannon's rate distortion: information in questions . . . . .	32
2.7	Discussion . . . . .	34
<b>3</b>	<b>Introducing MDL</b>	<b>37</b>
3.1	Introduction and overview . . . . .	37
3.2	The fundamental idea: learning as data compression . . . . .	38
3.2.1	Kolmogorov complexity and ideal MDL . . . . .	39
3.2.2	Practical MDL . . . . .	40
3.3	MDL and model selection . . . . .	40
3.4	Crude and refined MDL . . . . .	43
3.5	The MDL philosophy . . . . .	45
3.6	MDL and Occam's razor . . . . .	47
3.7	History . . . . .	49
3.8	Summary and outlook . . . . .	50
<b>4</b>	<b>Minimum description length</b>	<b>51</b>
4.1	Information theory I: probabilities and codelengths . . . . .	52
4.1.1	Prefix codes . . . . .	53
4.1.2	The Kraft inequality - codelengths and probabilities I . . . . .	53
4.1.3	The information inequality - codelengths and probabilities II . .	58
4.2	Statistical preliminaries and example models . . . . .	59
4.3	Crude MDL . . . . .	61
4.3.1	Description length of data given hypotheses . . . . .	61
4.3.2	Description length of hypotheses . . . . .	62
4.4	Information theory II: universal codes and models . . . . .	64
4.4.1	Two-part codes as simple universal codes . . . . .	66
4.4.2	From universal codes to universal models . . . . .	67
4.4.3	NML as an optimal universal model . . . . .	68
4.5	Simple refined MDL and its four interpretations . . . . .	71
4.5.1	Compression interpretation . . . . .	72



4.5.2	Counting interpretation . . . . .	72
4.5.3	Bayesian interpretation . . . . .	75
4.5.4	Prequential interpretation . . . . .	76
4.6	General refined MDL: gluing it all together . . . . .	79
4.6.1	Model selection with infinitely many models . . . . .	80
4.6.2	The infinity problem . . . . .	80
4.6.3	The general picture . . . . .	83
4.7	Beyond parametric model selection . . . . .	84
4.8	Relations to other approaches to inductive inference . . . . .	87
4.8.1	What is MDL? . . . . .	87
4.8.2	MDL and Bayesian inference . . . . .	88
4.8.3	MDL, prequential analysis and cross validation . . . . .	90
4.8.4	Kolmogorov complexity and structure function: ideal MDL . . . . .	91
4.9	Problems for MDL? . . . . .	91
4.9.1	Conceptual problems: Occam's razor . . . . .	91
4.9.2	Practical problems with MDL . . . . .	93
4.10	Discussion . . . . .	94
<b>5</b>	<b>Model uncertainty</b> . . . . .	<b>95</b>
5.1	Accounting for model uncertainty . . . . .	95
5.2	Bayesian framework and selection of prior distributions . . . . .	96
5.3	Model selection using Occam's window . . . . .	98
5.4	Markov chain Monte Carlo model composition . . . . .	100
5.5	Freedman's paradox resolved . . . . .	100
5.6	Discussion . . . . .	102
<b>6</b>	<b>Making predictions reliable</b> . . . . .	<b>105</b>
6.1	Entropification of a model class . . . . .	107
6.2	Entropification and MDL . . . . .	129
6.3	Discussion . . . . .	135
<b>7</b>	<b>Epilogue: using models in a careful way</b> . . . . .	<b>137</b>



## 1. Introduction

Signal processing is concerned with the representation, manipulation, and transformation of signals and the information that they carry. For example, we may wish to enhance a signal by reducing the noise or some other interference. We also may want to classify an object by means of its signal content. In order to classify we need an appropriate model. An essential constituent is the model length. The correct choice of the model length improves the use of the model in classification. Once it is possible to accurately model a signal, it then becomes possible to perform important signal processing tasks.

Conditioning on a single method ignores model uncertainty, and thus leads to the underestimation of uncertainty when making inferences about quantities of interest. A complete Bayesian solution to this problem involves averaging over all possible models when making such inferences. This approach is often not practical. An alternative approach involves averaging over a reduced set of models. Further, one can directly approximate the complete solution by applying a Markov chain Monte Carlo approach. In this approach the posterior distribution of a quantity of interest is approximated by a Markov chain Monte Carlo method which generates a process that moves through model space. This is discussed in chapter 5.

In order to assess the predictive ability of the selected models for future observations we must develop a measure of the effectiveness of a model selection strategy. A possible approach is to compare the quality of the predictions based on model averaging with the quality of predictions based on any single model. The choice of which procedure to use will depend on the particular application.

Reliable inferences allow one to make good predictions and decisions regarding the data under a much wider variety of assumptions than unreliable inferences do. It will allow us to establish in what way we can and in what way we cannot use overly simple models. In general, we will be interested in what can be reliably predicted - and what not - from a model that is only partially correct.

With an entropified model, if given enough data, we can find the model with the smallest expected prediction error. This model will provide a correct estimate of the average prediction error that it will achieve; hence the model gives a good impression of 'how good it really is' when errors are measured. Entropification is covered in chapter 6.

A detailed discussion concerning how the problematic issues are resolved is presented in the epilogue. We start with an introduction to make intuitive the theory dealing with the quantity of information in individual objects, continued by a non-technical introduction to the MDL principle. We conclude the introduction making precise the non-technicalities.



## 2. Kolmogorov complexity and information theory

How should we measure the amount of information about a phenomenon that is given to us by a particular observation concerning the phenomenon?

Shannon information theory, usually called just 'information' theory, was introduced in 1948 by C.E. Shannon (1916-2001). Kolmogorov complexity theory is also known as 'algorithmic information' theory. It was introduced independently and with different motivations by R.J. Solomonoff (born 1926), A.N. Kolmogorov (1903-1987) and G. Chaitin (born 1943) in 1960/1964, 1965 and 1966 respectively. Both theories aim at providing a means for measuring 'information'. They use the same unit to do this: the bit. In both cases, the amount of information in an object may be interpreted as the length of a description of the object. In the Shannon approach, however, the method of encoding objects is based on the presupposition that the objects to be encoded are outcomes of a known random source - it is only the characteristics of that random source that determine the encoding, not the characteristics of the objects that are its outcomes. In the Kolmogorov complexity approach we consider the individual objects themselves, in isolation so-to-speak, and the encoding of an object is a computer program (Turing machine) that generates it and then halts. In the Shannon approach we are interested in the minimum expected number of bits to transmit a message from a random source of known characteristics through an error-free channel. In Kolmogorov complexity we are interested in the minimum number of bits from which a particular message can effectively be reconstructed. A little reflection reveals that this is a great difference: for every source emitting but two messages the Shannon information is at most 1 bit, but we can choose both messages concerned of arbitrarily high Kolmogorov complexity. Shannon stresses in his founding article that his notion is only concerned with communication, while Kolmogorov stresses in his founding article that his notion aims at supplementing the gap left by Shannon theory concerning the information in individual objects. To be sure, both notions are natural: Shannon ignores the object itself but considers only the characteristics of the random source of which the object is one of the possible outcomes, while Kolmogorov considers only the object itself to determine the number of bits in the ultimate compressed version irrespective of the manner in which the object arose. Furthermore, we note that Shannon's approach is based on probability distributions and the approach of Kolmogorov dispenses with this notion.

In this chapter, we introduce, compare and contrast the Shannon and Kolmogorov approaches. We do this by switching back and forth between the two theories, according to the following pattern: we first discuss a concept of Shannon's theory, discuss its properties as well as some questions it leaves open. We then provide Kolmogorov's analogue of the concept and show how it answers the questions left open by Shannon's theory. We use as our guiding motif the communication between a sender A and a receiver B; where appropriate, we also discuss the related setting of a question-answer session between B and A.

To obtain an understanding of the two theories and how they relate, it is crucial to read the overview below and then section 2.2 and section 2.3, which discuss preliminaries,

fix notation and introduce the basic notions. The other sections are written in a way so that they can be read separately from one another. Throughout the chapter, we assume some basic familiarity with elementary notions of probability theory and computation.

The chapter does not contain any new results. All theorems that we present here, as well as further details, context and discussion, can be found in either of two standard text books: (Cover and Thomas, 1991, [20]), the standard reference on Shannon information theory, and/or (Li and Vitányi, 1997, [71]), the standard reference on Kolmogorov complexity.

## 2.1 Overview and summary

A summary of the basic ideas is given below. In the chapter, these notions are discussed in the same order.

1. Coding: Prefix codes, Kraft inequality. Since descriptions or encoding of objects are fundamental to both theories, we first review some elementary facts about coding. The most important of these is the Kraft inequality. This inequality gives the fundamental relationship between probability mass functions and prefix codes, which are the type of codes we are interested in (section 2.2).
2. Shannon's Fundamental Concept: Entropy is defined as a functional that maps probability distributions or, equivalently, random variables, to real numbers. This notion is derived from first principles as the only 'reasonable' way to measure the 'average amount of information conveyed when an outcome of the random variable is observed'. The notion is then related to encoding and communicating messages by Shannon's famous 'coding theorem' (section 2.3.1).
3. Kolmogorov's Fundamental Concept: Kolmogorov Complexity is defined as a function that maps objects (to be thought of as natural numbers or sequences of symbols) to the natural numbers. Intuitively, the Kolmogorov complexity of a sequence is the length (in bits) of the shortest computer program that prints the sequence and then halts (Section 2.3.2).
4. Universal Coding: interpolating between Shannon and Kolmogorov. Although their primary aim is quite different, and they are functions defined on different spaces, there are close relations between entropy and Kolmogorov complexity. These are best illustrated by explaining 'universal coding' which combines elements from both Shannon's and Kolmogorov's theory, and which lies at the basis of most practical data compression methods (Section 2.4).

Entropy and Kolmogorov Complexity are the basic notions of the two theories. They serve as building blocks for all other important notions in the respective theories. Arguably the most important of these notions is mutual information:

5. Mutual Information for Shannon and Kolmogorov: Entropy and Kolmogorov complexity are concerned with information in a single object: a random variable (Shannon) or an individual sequence (Kolmogorov). Both theories provide a (distinct) notion of mutual information that measures the information that one object gives about another object. In Shannon's theory, this is the information that one random variable carries about another; in Kolmogorov's theory ('algorithmic mutual information'), it is the information a sequence gives about another one (Section 2.5).

Entropy, Kolmogorov complexity and mutual information are concerned with lossless description or compression: messages must be described in such a way that from the description, the original message can be completely reconstructed. Extending the theories to lossy description or compression enables the formalization of more sophisticated concepts, such as ‘meaningful information’ and ‘useful information’. ‘Meaningful information’ is defined in the Kolmogorov framework using the Kolmogorov structure function. ‘Useful information’ is defined in Shannon’s framework using the rate-distortion function. We end the chapter with a brief treatment of the latter:

6. **Useful Information:** Rate-distortion theory is the part of Shannon information theory that deals with the following situation: The sender is only allowed to use a fixed (small) number of bits to send his message. The goal is then to send the most useful or valuable message given this constraint (Section 2.6).

## 2.2 The coding framework and the Kraft inequality

### Notational preliminaries:

1. **Strings** Let  $\mathcal{X}$  be some finite or countable set. We use the notation  $\mathcal{X}^*$  to denote the set of finite strings or sequences over  $\mathcal{X}$ . For example,

$$\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\},$$

with  $\epsilon$  denoting the empty word ‘’ with no letters. Let  $x, y, z \in \mathcal{N}$ , where  $\mathcal{N}$  denotes the natural numbers. We identify  $\mathcal{N}$  and  $\{0, 1\}^*$  according to the correspondence

$$(0, \epsilon), (1, 0), (2, 1), (3, 00), (4, 01), \dots \quad (2.1)$$

The length  $l(x)$  of  $x$  is the number of bits in the binary string  $x$ . For example,  $l(010) = 3$  and  $l(\epsilon) = 0$ . If  $x$  is interpreted as an integer, it can be shown that  $l(x) = \lfloor \log(x + 1) \rfloor$  and, for  $x \geq 2$ ,

$$\lfloor \log x \rfloor \leq l(x) \leq \lceil \log x \rceil. \quad (2.2)$$

Here, as in the sequel,  $\lceil x \rceil$  is the smallest integer larger than or equal to  $x$ ,  $\lfloor x \rfloor$  is the largest integer smaller than or equal to  $x$  and  $\log$  denotes logarithm to base two. We shall typically be concerned with encoding finite-length binary strings by other finite-length binary strings. The emphasis is on binary strings only for convenience; observations in any alphabet can be so encoded in a way that is ‘theory neutral’.

2. **(In)equality up to a constant** We will denote by  $\stackrel{+}{<}$  an inequality to within an additive constant. More precisely, let  $f, g$  be functions from  $\{0, 1\}^*$  to  $\mathbb{R}$ . Then by ‘ $f(x) \stackrel{+}{<} g(x)$ ’ we mean that there exists a  $c$  such that for all  $x \in \{0, 1\}^*$ ,  $f(x) < g(x) + c$ . We denote by  $\stackrel{\pm}{=}$  the situation when both  $\stackrel{+}{<}$  and  $\stackrel{+}{>}$  hold.
3. **Probabilities** Let  $P$  be a probability distribution defined on a finite or countable set  $\mathcal{X}$ . Throughout this chapter, we denote by  $X$  the random variable that takes values in  $\mathcal{X}$ , thus  $P(X = x) = P(\{x\})$  is the probability that the event  $\{x\}$  obtains. We write both  $P(x)$  and  $p_x$  as an abbreviation of  $P(X = x)$ .

**Codes** We repeatedly consider the following scenario: a sender (say, A) wants to communicate or transmit some information to a receiver (say, B). The information to be transmitted is an element from some set  $\mathcal{X}$ . It will be communicated by sending a binary string, called the message. When B receives the message, he can decode it again and (hopefully) reconstruct the element of  $\mathcal{X}$  that was sent. To achieve this, A and B need to agree on a code or description method before communicating. Intuitively, this is a binary relation between source words and associated code words. The relation is fully characterized by the decoding function. Such a decoding function  $D$  can be any function  $D : \{0, 1\}^* \rightarrow \mathcal{X}$ . The domain of  $D$  is the set of code words and the range of  $D$  is the set of source words.  $D(y) = x$  is interpreted as ‘ $y$  is a code word for the source word  $x$ ’. The set of all code words for source word  $x$  is the set  $D^{-1}(x) = \{y : D(y) = x\}$ . Hence,  $E = D^{-1}$  can be called the encoding substitution ( $E$  is not necessarily a function). With each code  $D$  we can associate a length function  $L_D : \mathcal{X} \rightarrow \mathbb{N}$  such that, for each source word  $x$ ,  $L(x)$  is the length of the shortest encoding of  $x$ :

$$L_D(x) = \min\{l(y) : D(y) = x\}.$$

We denote by  $x^*$  the shortest  $y$  such that  $D(y) = x$ ; if there is more than one such  $y$ , then  $x^*$  is defined to be the first such  $y$  in lexicographical order.

In coding theory attention is often restricted to the case where the source word set is finite, say  $X = \{1, 2, \dots, N\}$ . If there is a constant  $l_0$  such that  $l(y) = l_0$  for all code words  $y$  (equivalently,  $L(x) = l_0$  for all source words  $x$ ), then we call  $D$  a fixed-length code. It is easy to see that  $l_0 \geq \log N$ . For instance, in teletype transmissions the source has an alphabet of  $N = 32$  letters, consisting of the 26 letters in the Latin alphabet plus 6 special characters. Hence, we need  $l_0 = 5$  binary digits per source letter. In electronic computers we often use the fixed-length ASCII code with  $l_0 = 8$ .

**Prefix code** It is immediately clear that in general we cannot uniquely recover  $x$  and  $y$  from  $E(xy)$ . Let  $E$  be the identity mapping. Then we have  $E(00)E(00) = 0000 = E(0)E(000)$ . We now introduce prefix codes, which do not suffer from this defect. A binary string  $x$  is a proper prefix of a binary string  $y$  if we can write  $y = xz$  for  $z \neq \epsilon$ . A set  $\{x, y, \dots\} \subseteq \{0, 1\}^*$  is prefix-free if for any pair of distinct elements in the set neither is a proper prefix of the other. A function  $D : \{0, 1\}^* \rightarrow \mathcal{X}$  defines a prefix-code if its domain is prefix-free. In order to decode a code sequence of a prefix-code, we simply start at the beginning and decode one code word at a time. When we come to the end of a code word, we know it is the end, since no code word is the prefix of any other code word in a prefix-code. Suppose we encode each binary string  $x = x_1x_2 \dots x_n$  as

$$\bar{x} = \underbrace{11 \dots 1}_{n \text{ times}} 0 x_1 x_2 \dots x_n.$$

The resulting code is prefix because we can determine where the code word  $\bar{x}$  ends by reading it from left to right without backing up. Note  $l(\bar{x}) = 2n + 1$ ; thus, we have encoded strings in  $\{0, 1\}^*$  in a prefix manner at the price of doubling their length. We can get a much more efficient code by applying the construction above to the length  $l(x)$  of  $x$  rather than  $x$  itself: define  $x' = \bar{l(x)}x$ , where  $l(x)$  is interpreted as a binary string according to the correspondence (2.1). Then the code  $D'$  with  $D'(x') = x$  is a prefix code satisfying, for all  $x \in \{0, 1\}^*$ ,  $l(x') = n + 2 \log n + 1$  (here we ignore the ‘rounding error’ in (2.2)).  $D'$  is used throughout this chapter as a standard code



to encode natural numbers in a prefix free-manner; we call it the standard prefix-code for the natural numbers. We use  $L_{\mathcal{N}}(x)$  as notation for  $l(x')$ . When  $x$  is interpreted as a number (using the correspondence (2.1) and (2.2)), we see that  $L_{\mathcal{N}}(x) = \log x + 2 \log \log x + 1$ .

**Prefix codes and the Kraft inequality** Let  $\mathcal{X}$  be the set of natural numbers and consider the straightforward non-prefix representation (2.1). There are two elements of  $\mathcal{X}$  with a description of length 1, four with a description of length 2 and so on. However, for a prefix code  $D$  for the natural numbers there are less binary prefix code words of each length: if  $x$  is a prefix code word then no  $y = xz$  with  $z \neq \epsilon$  is a prefix code word. Asymptotically there are less prefix code words of length  $n$  than the  $2^n$  source words of length  $n$ . Quantification of this intuition for countable  $\mathcal{X}$  and arbitrary prefix-codes leads to a precise constraint on the number of code-words of given lengths. This important relation is known as the Kraft Inequality and is due to L.G. Kraft [20].

**Theorem 2.1 (Kraft inequality)**

*Let  $l_1, l_2, \dots$  be a finite or infinite sequence of natural numbers. There is a prefix-code with this sequence as lengths of its binary code words if*

$$\sum_n 2^{-l_n} \leq 1.$$

**Uniquely decodable codes** We want to code elements of  $\mathcal{X}$  in a way that they can be uniquely reconstructed from the encoding. Such codes are called ‘uniquely decodable’. Every prefix-code is a uniquely decodable code. For example, let  $\mathcal{X} = \{1, 2, 3, 4\}^*$ . If  $E(1) = 0$ ,  $E(2) = 10$ ,  $E(3) = 110$ ,  $E(4) = 111$  then 1421 is encoded as 0111100, which can be easily decoded from left to right in a unique way.

On the other hand, not every uniquely decodable code satisfies the prefix condition. Prefix-codes are distinguished from other uniquely decodable codes by the property that the end of a code word is always recognizable as such. This means that decoding can be accomplished without the delay of observing subsequent code words, which is why prefix-codes are also called instantaneous codes.

There is a good reason for our emphasis on prefix-codes. Namely, it turns out that theorem 2.1 stays valid if we replace ‘prefix-code’ by ‘uniquely decodable code’. This important fact means that every uniquely decodable code can be replaced by a prefix-code without changing the set of code-word lengths. In Shannon’s and Kolmogorov’s theories, we are only interested in code word lengths of uniquely decodable codes rather than the actual encoding. By the previous argument, we may restrict the set of codes we work with to prefix codes, which are much easier to handle.

**Probability distributions and complete prefix codes** A uniquely decodable code is complete if the addition of any new code word to its code word set results in a non-uniquely decodable code. It is easy to see that a code is complete if equality holds in the associated Kraft Inequality. Let  $l_1, l_2, \dots$  be the code words of some complete uniquely decodable code. Let us define  $q_x = 2^{-l_x}$ . By definition of completeness, we have  $\sum_x q_x = 1$ . Thus, the  $q_x$  can be thought of as probability mass functions corresponding to some probability distribution  $Q$ . We say  $Q$  is the distribution corresponding to  $l_1, l_2, \dots$ . In this way, each complete uniquely decodable code is mapped

to a unique probability distribution. Of course, this is nothing more than a formal correspondence: we may choose to encode outcomes of  $X$  using a code corresponding to a distribution  $Q$ , whereas the outcomes are actually distributed according to some  $P \notin Q$ . But, as we show in theorem 2.3 below, if  $X$  is distributed according to  $P$ , then the code to which  $P$  corresponds is, in an average sense, the code that achieves optimal compression of  $X$ .

**Prefix codes as protocols for asking questions** Prefix codes can be thought of as protocols for sequentially asking yes/no-questions. To make this precise we slightly change our setting. We now think of the ‘receiver’ B as someone who sequentially asks questions about  $X$ . We assume that the ‘sender’ A only passes on information when asked a question. But in that case, he answers truthfully. The questions of B must all be of the form ‘Is the realized value  $x$  an element of the set  $\mathcal{X}'$ ’, where  $\mathcal{X}'$  is some subset of  $\mathcal{X}$ . B keeps asking such questions until he has determined the precise value  $X = x$ . More precisely, B determines a sequence of sets  $\mathcal{X}_\epsilon, \mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_{00}, \mathcal{X}_{01}, \dots$ , satisfying the following two conditions:

1.  $\mathcal{X}_\epsilon = \mathcal{X}$ .
2. Let  $y \in \{0, 1\}^*$ . If  $\mathcal{X}_y$  has more than one element, then  $\mathcal{X}_{y0} \cap \mathcal{X}_{y1} = \emptyset$  and  $\mathcal{X}_{y0} \cup \mathcal{X}_{y1} = \mathcal{X}_y$ . If  $\mathcal{X}_y$  has just one element, then  $\mathcal{X}_{yz}$  is undefined for any continuation  $z$  of  $y$ .

The sets  $\mathcal{X}_y$  determine B’s protocol as follows. First, B asks ‘Is  $x \in \mathcal{X}_0$ ?’ If the answer is yes, then B’s next question is ‘Is  $x \in \mathcal{X}_{00}$ ?’ If the answer is no, then B knows that  $x \in \mathcal{X}_1$  and B’s next question is ‘Is  $x \in \mathcal{X}_{10}$ ?’ If the answer to the first two questions is yes, B’s third question is ‘Is  $x \in \mathcal{X}_{000}$ ?’ If the answer to the first question is no and to the second yes, then B’s question is ‘Is  $x \in \mathcal{X}_{10}$ ’, and so on. B keeps asking questions in this way until it has precisely determined the value of  $x$ , i.e. until it knows that  $x \in \mathcal{X}_y$  for some  $y$  such that  $\mathcal{X}_y$  has but one element.

To relate such a sequential protocol to prefix codes, consider the code  $E$  defined as follows: for all  $x \in \mathcal{X}$ , we set  $E(x) := y$  for the  $y$  such that  $\mathcal{X}_y = \{x\}$ . In this way all  $x \in \mathcal{X}$  are assigned a unique code word  $E(x)$  such that the set of code words is prefix-free. Therefore,  $E$  defines a prefix-code that, for each source word, reserves exactly one code word. Conversely, one can show that each prefix-code that reserves only one code word for each source word coincides with a sequential question-protocol.

Thus, the problems of prefix-free encoding the value of  $X$  and sequentially determining (by asking) the value of  $X$  are really equivalent. This is yet another reason why prefix codes are more ‘natural’ than general uniquely decodable codes.

## 2.3 Shannon entropy versus Kolmogorov complexity

### 2.3.1 Shannon entropy

It seldom happens that a detailed mathematical theory springs forth in essentially final form from a single publication. Such was the case with Shannon information theory, which properly started only with the appearance of C.E. Shannon’s paper ‘The mathematical theory of communication’ (Shannon, 1948, [102]). In this paper, Shannon proposed a measure of information in a distribution, which he called the ‘entropy’.

The entropy  $H(P)$  of a distribution  $P$  measures the ‘the inherent uncertainty in  $P$ ’, or (in fact equivalently), ‘how much information is gained when an outcome of  $P$  is observed’. To make this more precise, let us imagine an observer who knows that  $X$  is distributed according to  $P$ . The observer then observes  $X = x$ . The entropy of  $P$  stands for the ‘uncertainty of the observer about the outcome  $x$  before he observes it’. Now think of the observer as a ‘receiver’ who receives the message conveying the value of  $X$ . From this dual point of view, the entropy stands for

the average amount of information that the observer has gained after receiving a realized outcome  $x$  of the random variable  $X$ . (\*)

Below, we first give Shannon’s mathematical definition of entropy, and we then connect it to its intuitive meaning (\*).

### Definition 2.1

Let  $\mathcal{X}$  be a finite or countable set, let  $X$  be a random variable taking values in  $\mathcal{X}$  with distribution  $P$ . Then the (Shannon-) entropy of random variable  $X$  is given by

$$H(X) = - \sum_{x \in \mathcal{X}} p_x \log p_x, \quad (2.3)$$

Entropy is defined here as a functional mapping random variables to real numbers. In many texts, entropy is, essentially equivalently, defined as a map from distributions of random variables to the real numbers. Thus, by definition:  $H(P) := H(X) = - \sum_{x \in \mathcal{X}} p_x \log p_x$ .

**Motivation** Shannon’s definition can be motivated in several different ways. The two most important ones are the axiomatic approach and the coding interpretation. In this chapter we concentrate on the latter, but we first briefly sketch the former. The idea of the axiomatic approach is to postulate a small set of eminently reasonable conditions that any measure of information relative to a distribution should satisfy. One then shows that the only measure satisfying all the postulates is the Shannon entropy. We outline this approach for finite sources  $\mathcal{X} = \{1, \dots, N\}$ . We look for a function  $H$  that maps probability distributions on  $\mathcal{X}$  to real numbers. For given distribution  $P$ ,  $H(P)$  should measure ‘how much information is gained on average when an outcome is made available’. We can write  $H(P) = H(p_1, \dots, p_N)$  where  $p_i$  stands for the probability of  $i$ . Suppose we require that

1.  $H(p_1, \dots, p_N)$  is continuous in  $p_1, \dots, p_N$ .
2. If all the  $p_i$  are equal,  $p_i = 1/N$ , then  $H$  should be a monotonic increasing function of  $N$ . With equally likely events there is more choice, or uncertainty, when there are more possible events.
3. If a choice is broken down into two successive choices, the original  $H$  should be the weighted sum of the individual values of  $H$ . Rather than formalizing this condition, we will give a specific example. Suppose that  $\mathcal{X} = \{1, 2, 3\}$ , and  $p_1 = \frac{1}{2}$ ,  $p_2 = \frac{1}{3}$ ,  $p_3 = \frac{1}{6}$ . We can think of  $x \in \mathcal{X}$  as being generated in a two-stage process. First, an outcome in  $\mathcal{X}' = \{0, 1\}$  is generated according to a distribution  $P'$  with  $p'_0 = p'_1 = \frac{1}{2}$ . If  $x' = 1$ , we set  $x = 1$  and the process stops. If  $x' = 0$ , then outcome ‘2’ is generated with probability  $2/3$  and outcome ‘3’ with probability  $1/3$ , and the process stops. The final results

have the same probabilities as before. In this particular case we require that

$$H\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2}H\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{1}{2}H(1).$$

Thus, the entropy of  $P$  must be equal to entropy of the first step in the generation process, plus the weighted sum (weighted according to the probabilities in the first step) of the entropies of the second step in the generation process. As a special case, if  $\mathcal{X}$  is the  $n$ -fold product space of another space  $\mathcal{Y}$ ,  $X = (Y_1, \dots, Y_n)$  and the  $Y_i$  are all independently distributed according to  $P_Y$ , then  $H(P_X) = nH(P_Y)$ . For example, the total entropy of  $n$  independent tosses of a coin with bias  $p$  is  $nH(p, 1 - p)$ .

Remarkably, Shannon (1948) proved that

### Theorem 2.2

*The only  $H$  satisfying the three above assumptions is of the form  $H = -K \sum_{i=1}^N p_i \log p_i$ , with  $K$  a constant.*

Thus, requirements (1)-(3) lead us to the definition of entropy (2.3) given above up to an (unimportant) scaling factor. We shall give a concrete interpretation of this factor later on. Besides the defining characteristics (1)-(3), the function  $H$  has a few other properties that make it attractive as a measure of information. We mention:

4.  $H(p_1, \dots, p_N)$  is a concave function of the  $p_i$ .
5. For each  $N$ ,  $H$  achieves its unique maximum for the uniform distribution  $p_i = 1/N$ .
6.  $H(p_1, \dots, p_N)$  is zero if one of the  $p_i$  has value 1. Thus,  $H$  is zero if and only if we do not gain any information at all if we are told that the outcome is  $i$  (since we already knew  $i$  would take place with certainty).

We note that there do exist variations of ‘entropy’ which violate one or more of requirements (1)-(3); a good example is the family of Rényi entropies [20]. While such alternative notions of entropy are useful in their own, restricted context, Shannon’s original definition remains by far the most important.

**Coding interpretation** Immediately after stating theorem 2.2, Shannon continues [102], ‘this theorem, and the assumptions required for its proof, are in no way necessary for the present theory. It is given chiefly to provide a certain plausibility to some of our later definitions. The real justification of these definitions, however, will reside in their implications’.

Thus, in the spirit of Shannon, we will henceforth concentrate on a very concrete interpretation of entropy in terms of the length (number of bits) needed to encode outcomes in  $\mathcal{X}$ . This provides much clearer intuitions; it lies at the root of the many practical applications of information theory, and, most importantly for us, it simplifies the comparison to Kolmogorov complexity.

### Example 2.1

We start with an example. The entropy of a random variable  $X$  with equally likely outcomes in a finite sample space  $\mathcal{X}$  is given by  $H(X) = \log \mathcal{X}$ . By choosing a particular message  $x$  from  $\mathcal{X}$ , we remove the entropy from  $X$  by the assignment  $X :=$

$x$  and produce or transmit information  $I = \log \mathcal{X}$  by our selection of  $x$ . We show below that  $I = \log \mathcal{X}$  (or, to be more precise, the integer  $I' = \lceil \log \mathcal{X} \rceil$ ) can be interpreted as the number of bits needed to be transmitted from an (imagined) sender to an (imagined) receiver.  $\diamond$

We now connect entropy to minimum average code lengths. These are defined as follows:

### Definition 2.2

Let source words  $x \in \{0, 1\}^*$  be produced by a random variable  $X$  with probability  $P(x) = p_x$  for the event  $X = x$ . The characteristics of  $X$  are fixed. Now consider prefix codes  $D : \{0, 1\}^* \rightarrow \mathcal{N}$  with one code word per source word and denote the length of the code word for  $x$  by  $l_x$ . We want to minimize the expected number of bits we have to transmit for the given source  $X$  and choose a prefix code  $D$  that achieves this. In order to do so, we must minimize the average code-word length  $\bar{L}_D = \sum_x p_x l_x$ . We define the minimal average code word length as  $\bar{L} = \min\{\bar{L}_D : D \text{ is a prefix-code}\}$ . A prefix-code  $D$  such that  $\bar{L}_D = \bar{L}$  is called an optimal prefix-code with respect to prior probability  $P$  of the source words.

The (minimal) average code length of an (optimal) code does not depend on the details of the set of code words, but only on the set of code-word lengths. It is just the expected code-word length with respect to the given distribution. Shannon discovered that the minimal average code word length is about equal to the entropy of the source word set. This is known as the noiseless coding theorem. The adjective 'noiseless' emphasizes that we ignore the possibility of errors.

### Theorem 2.3

Let  $\bar{L}$  and  $P$  be as above. If  $H(P) = -\sum_x p_x \log p_x$  is the entropy, then

$$H(P) \leq \bar{L} \leq H(P) + 1. \quad (2.4)$$

We are typically interested in encoding a binary string of length  $n$  with entropy proportional to  $n$  (example 2.3). The essence of (2.4) is that, for all but the smallest  $n$ , the difference between entropy and minimal expected code length is completely negligible.

It turns out that the optimum  $\bar{L}$  in (2.4) is relatively easy to achieve, with the Shannon-Fano code. Let there be  $N$  symbols (also called basic messages or source words). Order these symbols according to decreasing probability, say  $\mathcal{X} = \{1, 2, \dots, N\}$  with probabilities  $p_1, p_2, \dots, p_N$ . Let  $P_r = \sum_{i=1}^{r-1} p_i$ , for  $r = 1, \dots, N$ . The binary code  $E : X \rightarrow \{0, 1\}^*$  is obtained by coding  $r$  as a binary number  $E(r)$ , obtained by truncating the binary expansion of  $P_r$  at length  $l(E(r))$  such that

$$-\log p_r \leq l(E(r)) < 1 - \log p_r.$$

This code is the Shannon-Fano code. It has the property that highly probable symbols are mapped to short code words and symbols with low probability are mapped to longer

code words (just like in a less optimal setting is done in the Morse code). Moreover,

$$2^{-l(E(r))} \leq p_r < 2^{-l(E(r))+1}.$$

Note that the code for symbol  $r$  differs from all codes of symbols  $r + 1$  through  $N$  in one or more bit positions, since for all  $i$  with  $r + 1 \leq i \leq N$ ,

$$P_i \geq P_r + 2^{-l(E(r))}.$$

Therefore the binary expansions of  $P_r$  and  $P_i$  differ in the first  $l(E(r))$  positions. This means that  $E$  is one-to-one, and it has an inverse: the decoding mapping  $E^{-1}$ . Even better, since no value of  $E$  is a prefix of any other value of  $E$ , the set of code words is a prefix-code. This means we can recover the source message from the code message by scanning it from left to right without look-ahead. If  $H_1$  is the average number of bits used per symbol of an original message, then  $H_1 = \sum_r p_r l(E(r))$ . Combining this with the previous inequality we obtain (2.4):

$$-\sum_r p_r \log p_r \leq H_1 < \sum_r (1 - \log p_r) p_r = 1 - \sum_r p_r \log p_r.$$

**Interpretation in terms of sequential questions** We re-interpret Shannon's noiseless coding theorem in terms of protocols for sequentially asking questions: suppose that B asks questions of the type 'Is  $x$  in the set  $\mathcal{X}'$ ?', where  $\mathcal{X}'$  is some subset of  $\mathcal{X}$ . A answers truthfully to each question, and B keeps asking questions until he has determined the exact value of the realized outcome  $x$  of the random variable  $X$ . In section 2.2 we showed that each protocol that B can use may be thought of as a prefix code with one code word per source word, and vice versa. Therefore, theorem 2.3 may be interpreted as follows. Suppose it is B's goal to determine the exact value of  $X$  using as few questions as possible. If B asks his questions in the cleverest possible way, he will on average need to ask  $H(X)$  questions (plus or minus one) to find out the exact value of  $X$ . From this point of view, the Shannon-Fano code we described above is a protocol for asking questions that is 'almost' optimal, where the 'optimal' protocol is the protocol that minimizes the expected number of questions to be asked.

**Problem and lacuna** Shannon observes, 'Messages have meaning [... however ...] the semantic aspects of communication are irrelevant to the engineering problem'. Thus, in Shannon's theory 'information' is fully determined by the probability distribution on the set of possible messages, and unrelated to the meaning, structure or content of individual messages. This is problematic in at least two ways:

First, in many practical cases, the distribution generating outcomes may be unknown to the observer or (worse), may not exist at all\*. For example, can we answer a question like 'what is the information in this book' by viewing it as an element of a set of possible books with a probability distribution on it? This seems unlikely. And how to measure the quantity of hereditary information in biological organisms, as encoded in DNA? Again there is the possibility of seeing a particular form of animal as one of a set of possible forms with a probability distribution on it. This seems to be contradicted by the fact that the calculation of all possible lifeforms in existence at any one time on earth would give a ridiculously low figure like  $2^{100}$ .

\* Even if we adopt a Bayesian (subjective) interpretation of probability, this problem remains [47].

Shannon's classical information theory assigns a quantity of information to an ensemble of possible messages. All messages in the ensemble being equally probable, this quantity is the number of bits needed to count all possibilities. This expresses the fact that each message in the ensemble can be communicated using this number of bits. However, it does not say anything about the number of bits needed to convey any individual message in the ensemble, and this constitutes a second 'lacuna' of Shannon's theory. To illustrate this, consider the ensemble consisting of all binary strings of length 9999999999999999. By Shannon's measure, we require 9999999999999999 bits on the average to encode a string in such an ensemble. However, the string consisting of 9999999999999999 1's can be encoded in about 55 bits by expressing 9999999999999999 in binary and adding the repeated pattern '1'. A requirement for this to work is that we have agreed on an algorithm that decodes the encoded string. We can compress the string still further when we note that 9999999999999999 equals  $3^2 \times 11111111111111$ , and that 11111111111111 consists of  $2^4$  1's.

Thus, we have discovered an interesting phenomenon: the description of some strings can be compressed considerably, provided they exhibit enough regularity. However, if regularity is lacking, it becomes more cumbersome to express large numbers. For instance, it seems easier to compress the number 'one billion', than the number 'one billion seven hundred thirty-five million two hundred sixty-eight thousand and three hundred ninety-four', even though they are of the same order of magnitude.

We are interested in a measure of information that, unlike Shannon's, does not rely on (often untenable) probabilistic assumptions, and that takes into account the phenomenon that 'regular' strings are compressible. Thus, we aim for a measure of information content of an individual finite object, and in the information conveyed about an individual finite object by another individual finite object. Here, we want the information content of an object  $x$  to be an attribute of  $x$  alone, and not to depend on, for instance, the means chosen to describe this information content. Surprisingly, this turns out to be possible, at least to a large extent. The resulting theory of information is based on Kolmogorov complexity, a notion independently proposed by Solomonoff [106], Kolmogorov [65] and Chaitin [16]; Li and Vitányi [71] describe the history of the subject.

### 2.3.2 Kolmogorov complexity

Suppose we want to describe a given object by a finite binary string. We do not care whether the object has many descriptions; however, each description should describe only one object. From among all descriptions of an object we can take the length of the shortest description as a measure of the object's complexity. It is natural to call an object 'simple' if it has at least one short description, and to call it 'complex' if all of its descriptions are long.

As in section 2.2, consider a description method  $D$ , to be used to transmit messages from a sender to a receiver. If  $D$  is known to both a sender and receiver, then a message  $x$  can be transmitted from sender to receiver by transmitting the description  $y$  with  $D(y) = x$ . The cost of this transmission is measured by  $l(y)$ , the length of  $y$ . The least cost of transmission of  $x$  is determined by the length function  $L(x)$ : recall that  $L(x)$  is the length of the shortest  $y$  such that  $D(y) = x$ . We choose this length function as the descriptonal complexity of  $x$  under specification method  $D$ .

Obviously, this descriptonal complexity of  $x$  depends crucially on  $D$ . The general principle involved is that the syntactic framework of the description language determines the succinctness of description.

In order to objectively compare descriptonal complexities of objects, to be able to say ' $x$  is more complex than  $z$ ', the descriptonal complexity of  $x$  should depend on  $x$  alone. This complexity can be viewed as related to a universal description method that is *a priori* assumed by all senders and receivers. This complexity is optimal if no other description method assigns a lower complexity to any object.

We are not really interested in optimality with respect to all description methods. For specifications to be useful at all it is necessary that the mapping from  $y$  to  $D(y)$  can be executed in an effective manner. That is, it can at least in principle be performed by humans or machines. This notion has been formalized as that of 'partial recursive functions', also known simply as computable functions (by Turing machines). According to generally accepted mathematical viewpoints it coincides with the intuitive notion of effective computation.

The set of partial recursive functions contains an optimal function that minimizes description length of every other such function. We denote this function by  $D_0$ . Namely, for any other recursive function  $D$ , for all objects  $x$ , there is a description  $y$  of  $x$  under  $D_0$  that is shorter than any description  $z$  of  $x$  under  $D$ . (That is, shorter up to an additive constant that is independent of  $x$ ). Complexity with respect to  $D_0$  minimizes the complexities with respect to all partial recursive functions.

We identify the length of the description of  $x$  with respect to a fixed specification function  $D_0$  with the 'algorithmic (descriptonal) complexity' of  $x$ . The optimality of  $D_0$  in the sense above means that the complexity of an object  $x$  is invariant (up to an additive constant independent of  $x$ ) under transition from one optimal specification function to another. Its complexity is an objective attribute of the described object alone: it is an intrinsic property of that object, and it does not depend on the description formalism. This complexity can be viewed as 'absolute information content': the amount of information that needs to be transmitted between all senders and receivers when they communicate the message in absence of any other *a priori* knowledge that restricts the domain of the message. Thus, we have outlined the program for a general theory of algorithmic complexity. The three major innovations are as follows:

1. In restricting ourselves to formally effective descriptions, our definition covers every form of description that is intuitively acceptable as being effective according to general viewpoints in mathematics and logic.
2. The restriction to effective descriptions entails that there is a universal description method that minimizes the description length or complexity with respect to any other effective description method. Significantly, this implies item 3.
3. The description length or complexity of an object is an intrinsic attribute of the object independent of the particular description method or formalizations thereof.

### 2.3.2.1 Formal details

The Kolmogorov complexity  $K(x)$  of a finite object  $x$  will be defined as the length of the shortest effective binary description of  $x$ . Broadly speaking,  $K(x)$  may be thought



of as the length of the shortest computer program that prints  $x$  and then halts. This computer program may be written in C, Java, LISP or any other universal language: we shall see that, for any two universal languages, the resulting program lengths differ at most by a constant not depending on  $x$ .

To make this precise, let  $T_1, T_2, \dots$  be a standard enumeration of all Turing machines, and let  $\phi_1, \phi_2, \dots$  be the enumeration of corresponding functions which are computed by the respective Turing machines. That is,  $T_i$  computes  $\phi_i$ . These functions are the partial recursive functions or computable functions. For technical reasons we are interested in the so-called prefix complexity, which is associated with Turing machines for which the set of programs (inputs) resulting in a halting computation is prefix free<sup>†</sup>. We can realize this by equipping the Turing machine with a one-way input tape, a separate work tape, and a one-way output tape. Such Turing machines are called prefix machines since the halting programs for any one of them form a prefix free set.

We first define  $K_{T_i}(x)$ , the prefix Kolmogorov complexity of  $x$  relative to a given prefix machine  $T_i$ , where  $T_i$  is the  $i$ -th prefix machine in a standard enumeration of them.  $K_{T_i}(x)$  is defined as the length of the shortest input sequence  $y$  such that  $T_i(y) = \phi_i(y) = x$ . If no such input sequence exists,  $K_{T_i}(x)$  remains undefined. Of course, this preliminary definition is still highly sensitive to the particular prefix machine  $T_i$  that we use. But now the 'universal prefix machine' comes to our rescue. Just as there exists universal ordinary Turing machines, there also exist universal prefix machines. These have the remarkable property that they can simulate every other prefix machine. More specifically, there exists a prefix machine  $U$  such that, with as input the pair  $\langle i, y \rangle$ , it outputs  $\phi_i(y)$  and then halts. We now fix, once and for all, a prefix machine  $U$  with this property and call  $U$  the reference machine. The Kolmogorov complexity  $K(x)$  of  $x$  is defined as  $K_U(x)$ .

Let us formalize this definition. Let  $\langle \cdot \rangle$  be a standard invertible effective one-one encoding from  $\mathcal{N} \times \mathcal{N}$  to a prefix-free subset of  $\mathcal{N}$ .  $\langle \cdot \rangle$  may be thought of as the encoding function of a prefix code. For example, we can set  $\langle x, y \rangle = x'y'$ .

We insist on prefix-freeness and recursiveness (i.e. partial recursive functions) because we want a universal Turing machine to be able to read an image under  $\langle \cdot \rangle$  from left to right and determine where it ends.

### Definition 2.3

Let  $U$  be our reference prefix machine, i.e. for all  $i \in \mathcal{N}$ ,  $y \in \{0, 1\}^*$ ,  $U(\langle i, y \rangle) = \phi_i(y)$ . The prefix Kolmogorov complexity of  $x$  is

$$\begin{aligned} K(x) &= \min_z \{l(z) : U(z) = x; z \in \{0, 1\}^*\} \\ &= \min_{i, y} \{l(\langle i, y \rangle) : \phi_i(y) = x, y \in \{0, 1\}^*, i \in \mathcal{N}\}. \end{aligned} \quad (2.5)$$

We can alternatively think of  $z$  as a program that prints  $x$  and then halts, or as  $z = \langle i, y \rangle$  where  $y$  is a program such that, when  $T_i$  is input program  $y$ , it prints  $x$  and then halts.

Thus, by definition  $K(x) = l(x^*)$ , where  $x^*$  is the lexicographically first shortest self-delimiting (prefix) program for  $x$  with respect to the reference prefix machine.

<sup>†</sup> There exists a version of Kolmogorov complexity corresponding to programs that are not necessarily prefix-free, but we will not go into it here.

Consider the mapping  $E^*$  defined by  $E^*(x) = x^*$ . This may be viewed as the encoding function of a prefix-code (decoding function)  $D^*$  with  $D^*(x^*) = x$ . By its definition,  $D^*$  is a very parsimonious code. The reason for working with prefix rather than standard Turing machines is that, for many of the subsequent developments, we need  $D^*$  to be prefix.

Though defined in terms of a particular machine model, the Kolmogorov complexity is machine-independent up to an additive constant and acquires an asymptotically universal and absolute character through Church's thesis ([71], p. 29), from the ability of universal machines to simulate one another and execute any effective process. The Kolmogorov complexity of an object can be viewed as an absolute and objective quantification of the amount of information in it.

### Example 2.2

To develop some intuitions, it is useful to think of  $K(x)$  as the shortest program for  $x$  in some standard programming language such as LISP or Java. Consider the lexicographical enumeration of all syntactically correct LISP programs  $\lambda_1, \lambda_2, \dots$ , and the lexicographical enumeration of all syntactically correct Java programs  $\pi_1, \pi_2, \dots$ . We assume that both these programs are encoded in some standard prefix-free manner. With proper definitions we can view the programs in both enumerations as computing partial recursive functions from their inputs to their outputs. Choosing reference machines in both enumerations we can define complexities  $K_{\text{LISP}}(x)$  and  $K_{\text{Java}}(x)$  completely analogous to  $K(x)$ . All of these measures of the descriptonal complexities of  $x$  coincide up to a fixed additive constant. Let us show this directly for  $K_{\text{LISP}}(x)$  and  $K_{\text{Java}}(x)$ . Since LISP is universal, there exists a LISP program  $\lambda_P$  implementing a Java-to-LISP compiler.  $\lambda_P$  translates each Java program to an equivalent LISP program. Consequently, for all  $x$ ,  $K_{\text{LISP}}(x) \leq K_{\text{Java}}(x) + 2l(P)$ . Similarly, there is a Java program  $\pi_L$  that is a LISP-to-Java compiler, so that for all  $x$ ,  $K_{\text{Java}}(x) \leq K_{\text{LISP}}(x) + 2l(L)$ . It follows that  $|K_{\text{Java}}(x) - K_{\text{LISP}}(x)| \leq 2l(P) + 2l(L)$  for all  $x$ !

The programming language view immediately tells us that  $K(x)$  must be small for 'simple' or 'regular' objects  $x$ . For example, there exists a fixed-size program that, when input  $n$ , outputs the first  $n$  bits of  $\pi$  and then halts. Specification of  $n$  takes at most  $L_N(n) = \log n + 2 \log \log n + 1$  bits. Thus, if  $x$  consists of the first  $n$  binary digits of  $\pi$ , then  $K(x) \stackrel{+}{\leq} \log n + 2 \log \log n$ . Similarly, if  $0^n$  denotes the string consisting of  $n$  0's, then  $K(0^n) \stackrel{+}{\leq} \log n + 2 \log \log n$ .

On the other hand, for all  $x$ , there exists a program 'print  $x$ ; halt'. This shows that for all  $K(x) \stackrel{+}{\leq} l(x)$ . As was previously noted, for any prefix code, there are no more than  $2^m$  strings  $x$  which can be described by  $m$  or less bits. In particular, this holds for the prefix code  $E^*$  whose length function is  $K(x)$ . Thus, the fraction of strings  $x$  of length  $n$  with  $K(x) \leq m$  is at most  $2^{m-n}$ : the overwhelming majority of sequences cannot be compressed by more than a constant. Specifically, if  $x$  is determined by  $n$  independent tosses of a fair coin, then with overwhelming probability,  $K(x) \approx l(x)$ . Thus, while for very regular strings, the Kolmogorov complexity is small (sublinear in the length of the string), most strings are 'random' and have Kolmogorov complexity about equal to their own length.  $\diamond$

**Problem and lacuna** Unfortunately  $K(x)$  is not a recursive function: the Kolmogorov complexity is not computable in general. This means that there exists no computer program that, when input an arbitrary string, outputs the Kolmogorov complexity of that string and then halts. This follows from Gödel's theorem of incompleteness [71]. While there exist 'feasible', resource-bounded forms of Kolmogorov complexity [71], these lack some of the elegant properties of the original, uncomputable notion.

Now suppose we are interested in efficient storage and transmission of long sequences of data. According to Kolmogorov, we can compress such sequences in an essentially optimal way by storing or transmitting the shortest program that generates them. Unfortunately, as we have just seen, we cannot find such a program in general. According to Shannon, we can compress such sequences optimally in an average sense (and therefore, it turns out, also with high probability) if they are distributed according to some  $P$  and we know  $P$ . Unfortunately, in practice,  $P$  is often unknown or even nonexistent. Thus, both Shannon's and Kolmogorov's idea are not directly applicable to most actual data compression problems. For these, we can use universal codes which may be viewed at the same time as an extension of Shannon's, and a 'downscaling' of Kolmogorov's theory.

## 2.4 Universal coding: interpolating between Kolmogorov and Shannon

Below we repeatedly use the coding concepts introduced in section 2.2. Suppose we are given a recursive enumeration of prefix codes  $D_1, D_2, \dots$ . Let  $L_1, L_2, \dots$  be the length functions associated with these codes. That is,  $L_i(x) = \min\{l(y) : D_i(y) = x\}$ ; if there exists no  $y$  with  $D_i(y) = x$ , then  $L_i(y) = 1$ . We may encode  $x$  by first encoding a natural number  $k$  using the standard prefix code for the natural numbers. We then encode  $x$  itself using the code  $D_k$ . This leads to a so-called two-part code  $\tilde{D}$  with lengths  $\tilde{L}$ . By construction, this code is prefix and its lengths satisfy

$$\tilde{L}(x) := \min_{k \in \mathcal{N}} L_{\mathcal{N}}(k) + L_k(x). \quad (2.6)$$

Let  $x$  be an infinite binary sequence and let  $x_{[1:n]} \in \{0, 1\}^n$  be the initial  $n$ -bit segment of this sequence. Since  $L_{\mathcal{N}}(k) = \mathcal{O}(\log k)$ , we have for all  $k$ , all  $n$ :

$$\tilde{L}(x_{[1:n]}) \leq L_k(x_{[1:n]}) + \mathcal{O}(\log k).$$

Recall that for each fixed  $L_k$ , the fraction of sequences of length  $n$  that can be compressed by more than  $m$  bits is less than  $2^{-m}$ . Thus, typically, the codes  $L_k$  and the strings  $x_{[1:n]}$  will be such that  $L_k(x_{[1:n]})$  grows linearly with  $n$ . This implies that for every  $x$ , the newly constructed  $\tilde{L}$  is 'almost as good' as whatever code  $D_k$  in the list is best for that particular  $x$ : the difference in code lengths is bounded by a constant depending on  $k$  but not on  $n$ . In particular, for each  $k$  and each infinite sequence  $x$ ,

$$\lim_{n \rightarrow \infty} \frac{\tilde{L}(x_{[1:n]})}{L_k(x_{[1:n]})} \leq 1. \quad (2.7)$$

A code satisfying (2.7) is called a universal code relative to the comparison class of codes  $\{D_1, D_2, \dots\}$ . It is 'universal' in the sense that it compresses every sequence

essentially as well as the  $D_k$  that compresses that particular sequence the most. In general, there exist many types of codes that are universal: the 2-part universal code defined above is just one means of achieving (2.7).

**Universal codes and Kolmogorov** In most practically interesting cases we may assume that for all  $k$ , the decoding function  $D_k$  is computable, i.e. there exists a prefix Turing machine which for all  $y \in \{0, 1\}^*$ , when input  $y'$  (the prefix-free version of  $y$ ), outputs  $D_k(y)$  and then halts. Since such a program has finite length, we must have for all  $k$ ,

$$l(E^*(x_{[1:n]})) = K(x_{[1:n]}) \stackrel{+}{\leq} L_k(x_{[1:n]}),$$

where  $E^*$  is the encoding function defined earlier, with  $l(E^*(x)) = K(x)$ . Comparing with (2.7) shows that the code  $D^*$  with encoding function  $E^*$  is a universal code relative to  $D_1, D_2, \dots$ . Thus, we see that the Kolmogorov complexity  $K$  is just the length function of the universal code  $D^*$ . Note that  $D^*$  is an example of a universal code that is not (explicitly) two-part.

### Example 2.3

Let us create a universal two-part code that allows us to significantly compress all binary strings with frequency of 0's deviating significantly from  $\frac{1}{2}$ . For  $n_0 < n_1$ , let  $D_{\langle n, n_0 \rangle}$  be the code that assigns code words of equal (minimum) length to all strings of length  $n$  with  $n_0$  zeroes, and no code words to any other strings. Then  $D_{\langle n, n_0 \rangle}$  is a prefix-code and  $L_{\langle n, n_0 \rangle}(x) = \lceil \log \binom{n}{n_0} \rceil$ . The universal two part code  $\tilde{D}$  relative to the set of codes  $\{D_{\langle i, j \rangle} : i, j \in \mathcal{N}\}$  then achieves the following lengths (to within 1 bit): for all  $n$ , all  $n_0 \in \{0, \dots, n\}$ , all  $x_{[1:n]}$  with  $n_0$  zeroes,

$$\begin{aligned} \tilde{L}(x_{[1:n]}) &= \log n + \log n_0 + 2 \log \log n + 2 \log \log n_0 + \log \binom{n}{n_0} \\ &= \log \binom{n}{n_0} + \mathcal{O}(\log n). \end{aligned} \quad (2.8)$$

Using Stirling's approximation of the factorial,  $n! \sim n^n \exp(-n) \sqrt{2\pi n}$ , we find that

$$\begin{aligned} \log \binom{n}{n_0} &= \log n! - \log n_0! - \log(n - n_0)! \\ &= n \log n - n_0 \log n_0 - (n - n_0) \log(n - n_0) + \mathcal{O}(\log n) \\ &= nH(n_0/n) + \mathcal{O}(\log n). \end{aligned} \quad (2.9)$$

Note that  $H(n_0/n) \leq 1$ , with equality if  $n_0 = n$ . Therefore, if the frequency deviates significantly from  $\frac{1}{2}$ ,  $\tilde{D}$  compresses  $x_{[1:n]}$  by a factor linear in  $n$ . In all such cases,  $D^*$  compresses the data by at least the same linear factor. Note that (a) each individual code  $D_{\langle n, n_0 \rangle}$  is capable of exploiting a particular type of regularity in a sequence to compress that sequence, (b) the universal code  $\tilde{D}$  may exploit many different types of regularities to compress a sequence, and (c) the code  $D^*$  with lengths given by the Kolmogorov complexity asymptotically exploits all computable regularities so as to maximally compress a sequence.  $\diamond$

**Universal codes and Shannon** If  $X$  is distributed according to some distribution  $P$ , then the optimal (in the average sense) code to use is the Shannon-Fano code. But now

suppose it is only known that  $P \in \mathcal{P}$ , where  $\mathcal{P}$  is some given (possibly very large, e.g. uncountable) set of candidate distributions. Now it is not clear what code is optimal. We may try the Shannon-Fano code for a particular  $P \in \mathcal{P}$ , but such a code will typically lead to very large expected code lengths if  $X$  turns out to be distributed according to some  $Q \in \mathcal{P}$ ,  $Q \neq P$ . We may ask whether there exists another code that is 'almost' as good as the Shannon-Fano code for  $P$ , no matter what  $P \in \mathcal{P}$  actually generates the sequence? We now show that, provided  $\mathcal{P}$  is finite or countable, then (perhaps surprisingly), the answer is yes. To see this, we need the notion of an information source. An information source may be thought of as a probability distribution over arbitrarily long sequences, of which an observer gets to see longer and longer initial segments; examples are given below. Formally, an information source  $P$  is a probability distribution on the set  $\{0, 1\}^\infty$  of one-way infinite sequences. Such a  $P$  can be identified with the distributions  $P^{(1)}$  on  $\{0, 1\}^1$ ,  $P^{(2)}$  on  $\{0, 1\}^2$ ,  $\dots$ . Here  $P^{(n)}$  denotes the marginal distribution of  $P$  on the first  $n$ -bit segments.  $P^{(n)}$  is related to  $P^{(n+1)}$  as follows: for all  $n \geq 0$ , all  $x \in \{0, 1\}^n$ ,  $\sum_{y \in \{0, 1\}} P^{(n+1)}(xy) = P^{(n)}(x)$  and  $P^{(0)}(x) = 1$ .

Suppose then that  $\mathcal{P}$  is a finite or countable set of information sources. Then the members of  $\mathcal{P}$  may be listed as  $P_1, P_2, \dots$ . To each marginal distribution  $P_k^{(n)}$ , there corresponds a unique Shannon-Fano code defined on the set  $\{0, 1\}^n$  with lengths  $L_{\langle n, k \rangle}(x) := \lceil -\log P_k^{(n)}(x) \rceil$ .

For given  $P \in \mathcal{P}$ , we define

$$H(P^{(n)}) := \sum_{x \in \{0, 1\}^n} P^{(n)}(x) \lceil -\log P^{(n)}(x) \rceil,$$

as the entropy of the distribution of the first  $n$  outcomes.

Let  $E$  be a prefix-code assigning code word  $E(x)$  to source word  $x \in \{0, 1\}^n$ . The noiseless coding theorem 2.3 asserts that the minimal average code word length  $\bar{L}(P) = \sum_{x \in \{0, 1\}^n} P(x) l(E(x))$  among all such prefix-codes  $E$  satisfies

$$H(P^{(n)}) \leq L(P) \leq H(P^{(n)}) + 1.$$

The entropy  $H(P^{(n)})$  can therefore be interpreted as the expected code length of encoding the first  $n$  bits generated by the source  $P$ , when the optimal (Shannon-Fano) code is used. We look for a prefix code  $\tilde{D}$  with length function  $\tilde{L}$  that satisfies, for all  $P \in \mathcal{P}$ :

$$\lim_{n \rightarrow \infty} \frac{E_P \tilde{L}(X_{[1:n]})}{H(P^{(n)})} \leq 1,$$

where  $E_P \tilde{L}(X_{[1:n]}) = \sum_{x \in \{0, 1\}^n} P^{(n)}(x) \tilde{L}(x)$ . Define  $\tilde{D}$  as the following two-part code: first,  $n$  is encoded using the standard prefix code for natural numbers. Then, among all codes  $D_{\langle n, k \rangle}$ , the  $k$  that minimizes  $L_{\langle n, k \rangle}(x)$  is encoded (again using the standard prefix code); finally,  $x$  is encoded in  $L_{\langle n, k \rangle}(x)$  bits. Then for all  $n$ , for all  $k$ , for every sequence  $x_{[1:n]}$ ,

$$\tilde{L}(x_{[1:n]}) \leq L_{\langle n, k \rangle}(x_{[1:n]}) + L_{\mathcal{N}}(k) + L_{\mathcal{N}}(n). \quad (2.10)$$

Since (2.10) holds for all strings of length  $n$ , it must also hold in expectation for all possible distributions on strings of length  $n$ . In particular, this gives, for all  $k \in \mathcal{N}$ ,

$$E_{P_k} \tilde{L}(X_{[1:n]}) \leq E_{P_k} L_{\langle n, k \rangle}(X_{[1:n]}) + \mathcal{O}(\log n) = H(P_k^{(n)}) + \mathcal{O}(\log n),$$

from which (2.4) follows.

Historically, codes satisfying (2.4) have been called universal codes relative to  $\mathcal{P}$ ; codes satisfying (2.7) have been considered in the literature only much more recently and are usually called ‘universal codes for individual sequences’ [77]. The two-part code  $\tilde{D}$  that we just defined is universal both in an individual sequence and in an average sense:  $\tilde{D}$  achieves code lengths within a constant of that achieved by  $D_{\langle n, k \rangle}$  for every individual sequence, for every  $k \in \mathcal{N}$ ; but  $\tilde{D}$  also achieves expected code lengths within a constant of the Shannon-Fano code for  $P$ , for every  $P \in \mathcal{P}$ . We may say that  $\tilde{D}$  interpolates between Shannon’s codes, which are optimal for a specific  $P$ , and Kolmogorov’s code  $D^*$  (with length function  $K$ ), which by definition does at least as well (within an additive constant) as  $\tilde{D}$ .

#### Example 2.4

Suppose our sequence is generated by independent tosses of a coin with bias  $p$  of tossing ‘head’ where  $p \in (0, 1)$ . Identifying ‘heads’ with 1, the probability of  $n - n_0$  outcomes ‘1’ in an initial segment  $x_{[1:n]}$  is then  $(1 - p)^{n_0} p^{n - n_0}$ . Let  $\mathcal{P}$  be the set of corresponding information sources, containing one element for each  $p \in (0, 1)$ .  $\mathcal{P}$  is an uncountable set; nevertheless, a universal code for  $\mathcal{P}$  exists. In fact, it can be shown that the code  $\tilde{D}$  with lengths (2.10) in example 2.3 is universal for  $\mathcal{P}$ , i.e. it satisfies (2.4). The reason for this is (roughly) as follows: if data are generated by a coin with bias  $p$ , then with probability 1, the frequency  $n_0/n$  converges to  $p$ , so that, by (2.10),  $n^{-1} \tilde{L}(x_{[1:n]})$  tends to  $n^{-1} H(P^{(n)}) = H(p, 1 - p)$ .

If we are interested in practical data-compression, then the assumption that the data are generated by a biased-coin source is very restricted. But there are much richer classes of distributions  $\mathcal{P}$  for which we can formulate universal codes. For example, we can take  $\mathcal{P}$  to be the class of all Markov sources of each order; here the probability that  $X_i = 1$  may depend on arbitrarily many earlier outcomes. Such ideas form the basis of most data compression schemes used in practice. Codes which are universal for the class of all Markov sources of each order and which encode and decode in real-time can easily be implemented. Thus, while we cannot find the shortest program that generates a particular sequence, it is often possible to effectively find the shortest encoding within a quite sophisticated class of codes.  $\diamond$

**Expected Kolmogorov complexity  $\pm$  Shannon entropy** Suppose the source words  $x$  are distributed as a random variable  $X$  with probability  $P(x)$ . While  $K(x)$  is fixed for each  $x$  and gives the shortest code word length (but only up to a fixed constant) and is independent of the probability distribution  $P$ , we may wonder whether  $K$  is also universal in the following sense: If we weigh each individual code word length for  $x$  with its probability  $P(x)$ , thus the resulting  $P$ -expected code word length  $\sum_x P(x) K(x)$  achieve the minimal average code word length  $H(P) = - \sum_x P(x) \log P(x)$ ? Here we sum over the entire support of  $P$ ; restricting summation to a small set, for example the singleton set  $\{x\}$ , can give a different result. The reasoning above implies that, under some mild restrictions on the distributions  $P$ , the answer is yes. This is expressed in the following theorem, where, instead of the quotient we look at the difference of  $\sum_x P(x) K(x)$  and  $H(P)$ . This allows us to express really small distinctions. We call an information source  $P$  recursive if there exists a Turing machine that, when input

$\langle n, x, y \rangle$  with  $x \in \{0, 1\}^*$  and  $y, n \in \mathcal{N}$ , outputs  $P^{(n)}(x)$  to precision  $\frac{1}{y}$ . The following theorem can be found in [71].

#### Theorem 2.4

Let  $P$  be a recursive information source. Then for all  $n$ ,

$$0 \leq \sum_{x \in \{0,1\}^n} P^{(n)}(x) K(x) - H(P^{(n)}) \leq c_P,$$

where  $c_P$  is a constant that depends only on  $P$  (and not on  $n$ ).

The Shannon-Fano code for a computable distribution is itself computable. Therefore, for every computable distribution  $P$ , the universal code  $D^*$  whose length function is the Kolmogorov complexity compresses on average at least as much as the Shannon-Fano code for  $P$ . This is the intuitive reason why, no matter what computable distribution  $P$  we take, its expected Kolmogorov complexity is close to its entropy.

## 2.5 Mutual information

### 2.5.1 Shannon mutual information

How much information can a random variable  $X$  convey about a random variable  $Y$ ? Taking a purely combinatorial approach, this notion is captured as follows. If  $X$  ranges over  $S_X$  and  $Y$  ranges over  $S_Y$ , then we look at the set  $U$  of possible events  $(X = a, Y = b)$  consisting of joint occurrences of event  $X = a$  and event  $Y = b$ . If  $U$  does not equal the Cartesian product  $S_X \times S_Y$ , then this means there is some dependency between  $X$  and  $Y$ . Considering the set  $U_a = \{(a, y) : (a, y) \in U\}$  for  $a \in S_X$ , it is natural to define the conditional entropy of  $Y$  given  $X = a$  as  $H(Y|X = a) = \log d(U_a)$ . This suggests immediately that the information given by  $X = a$  about  $Y$  is

$$I(X = a : Y) = H(Y) - H(Y|X = a).$$

For example, if  $U = \{(1, 1), (1, 2), (2, 3)\}$ ,  $U \subseteq S_X \times S_Y$  with  $S_X = \{1, 2\}$  and  $S_Y = \{1, 2, 3, 4\}$ , then  $I(X = 1 : Y) = 1$  and  $I(X = 2 : Y) = 2$ .

In this formulation it is obvious that  $H(X|X = a) = 0$ , and that  $I(X = a : X) = H(X)$ . This approach amounts to the assumption of uniform distribution of the probabilities concerned. We can generalize this approach, taking into account the frequencies or probabilities of the occurrences of the different values  $X$  and  $Y$  can assume. Let the joint probability  $p(a, b)$  be defined as: 'the probability of the joint occurrence of event  $X = a$  and event  $Y = b$ '. This leads to the self-evident formulas for joint variables  $X, Y$ :

$$\begin{aligned} H(X, Y) &= - \sum_{a,b} p(a, b) \log p(a, b), \\ H(X) &= - \sum_{a,b} p(a, b) \log \sum_b p(a, b), \\ H(Y) &= - \sum_{a,b} p(a, b) \log \sum_a p(a, b), \end{aligned}$$

where summation over  $a$  is taken over all outcomes of the random variable  $X$  and summation over  $b$  is taken over all outcomes of random variable  $Y$ . One can show that

$$H(X, Y) \leq H(X) + H(Y), \quad (2.11)$$

with equality only in the case that  $X$  and  $Y$  are independent. In all of these equations the entropy quantity on the left-hand side increases if we choose the probabilities on the right-hand side more equally.

**Conditional entropy** The conditional probability  $p(b|a)$  of outcome  $Y = b$  given outcome  $X = a$  for random variables  $X$  and  $Y$  (not necessarily independent) is defined by

$$p(b|a) = \frac{p(a, b)}{\sum_b p(a, b)},$$

This leads to the following analysis of the information in  $X$  about  $Y$ ; by first considering the conditional entropy of  $Y$  given  $X$  as the average of the entropy for  $Y$  for each value of  $X$  weighted by the probability of getting that particular value:

$$\begin{aligned} H(Y|X) &= \sum_a p(a) H(Y|X = a) \\ &= - \sum_a p(a) \sum_b p(b|a) \log p(b|a) \\ &= - \sum_{a,b} p(a, b) \log p(b|a). \end{aligned}$$

The quantity on the left-hand side tells us how uncertain we are about the outcome of  $Y$  when we know an outcome of  $X$ . With

$$\begin{aligned} H(X) &= - \sum_a p(a) \log p(a) \\ &= - \sum_a \left( \sum_b p(a, b) \right) \log \sum_b p(a, b) \\ &= - \sum_{a,b} p(a, b) \log \sum_b p(a, b), \end{aligned}$$

and substituting the formula for  $p(b|a)$ , we find  $H(Y|X) = H(X, Y) - H(X)$ . Rewrite this expression as the entropy equality

$$H(X, Y) = H(X) + H(Y|X). \quad (2.12)$$

This can be interpreted as, ‘the uncertainty of the joint event  $(X, Y)$  is the uncertainty of  $X$  plus the uncertainty of  $Y$  given  $X$ ’. Combining (2.11) and (2.12) gives  $H(Y) \geq H(Y|X)$ , which can be taken to imply that knowledge of  $X$  can never increase uncertainty of  $Y$ . In fact, uncertainty in  $Y$  will be decreased unless  $X$  and  $Y$  are independent. Finally, the information in the outcome  $X = a$  about  $Y$  is defined as

$$I(X = a : Y) = H(Y) - H(Y|X = a). \quad (2.13)$$

Here the quantities  $H(Y)$  and  $H(Y|X = a)$  on the right-hand side of the equations are always equal to or less than the corresponding quantities under the uniform distribution



we analyzed first. The values of the quantities  $I(X = a : Y)$  under the assumption of uniform distribution of  $Y$  and  $Y|X = a$  versus any other distribution are not related by inequality in a particular direction. The equalities  $H(X|X = a) = 0$  and  $I(X = a : X) = H(X)$  hold under any distribution of the variables. Since  $I(X = a : Y)$  is a function of outcomes of  $X$ , while  $I(Y = b : X)$  is a function of outcomes of  $Y$ , we do not compare them directly. However, forming the expectation defined as

$$E(I(X = a : Y)) = \sum_a p(a)I(X = a : Y),$$

$$E(I(Y = b : X)) = \sum_b p(b)I(Y = b : X),$$

and combining (2.12) and (2.13), we see that the resulting quantities are equal. Denoting this quantity by  $I(X, Y)$  and calling it the mutual information in  $X$  and  $Y$ , we see that this information is symmetric:

$$I(X, Y) = E(I(X = a : Y)) = E(I(Y = b : X)). \quad (2.14)$$

### Example 2.5

Suppose we want to exchange the information about the outcome  $X = x$  and it is known already that outcome  $Y = y$  is the case. Then we require (using the Shannon-Fano code) about  $\log P(X = x|Y = y)$  bits to communicate  $x$ . On average, over the joint distribution  $P(X = x, Y = y)$  we use  $H(X|Y)$  bits, which is optimal by Shannon's noiseless coding theorem. In fact, exploiting the mutual information paradigm, the expected information that outcome  $Y = y$  gives about outcome  $X = x$  is the same as the expected information that  $X = x$  gives about  $Y = y$ .  $\diamond$

**Interpretation in terms of sequential questions** Just as we did for the entropy, we can also re-interpret mutual information in terms of protocols for asking questions. Suppose that B sequentially asks questions about  $X$ , but, as in example 2.5, before he has to ask any questions, B is told that  $Y = y$ . B then sequentially asks questions to find out the value of  $X$ , using the protocol defined by the Shannon-Fano code for  $P(X = \cdot | Y = y)$ . By Shannon's noiseless coding theorem, this is the optimal protocol. Intuitively, since B is given some initial information, we expect that B has to ask fewer questions than if he were not given any initial information.  $I(Y; X)$  denotes exactly how many fewer questions B can expect to need to ask on average if he is already told the value of  $Y$  before asking any questions. Here the average is over both  $X$  and  $Y$ . Indeed, on average, B needs to ask fewer questions, since  $I(Y; X) \geq 0$ . But there may certainly exist individual  $y$  such that  $I(Y = y : X)$  is negative. For example, we may have  $\mathcal{X} = \{0, 1\}$ ,  $\mathcal{Y} = \{0, 1\}$ ,  $P(X = 1|Y = 0) = 1$ ,  $P(X = 1|Y = 1) = \frac{1}{2}$ ,  $P(Y = 1) = \epsilon$ . Then  $I(Y; X) = H(\epsilon, 1 - \epsilon)$  whereas  $I(Y = 1 : X) = H(\epsilon, 1 - \epsilon) + \epsilon - 1$ . For small  $\epsilon$ , this quantity is smaller than 0.

**Problem and lacuna** The quantity  $I(X; Y)$  symmetrically characterizes to what extent random variables  $X$  and  $Y$  are correlated. An inherent problem with probabilistic definitions is that - as we have just seen - although  $E(I(Y : X))$  is always positive, for some probability distributions and some  $y$ ,  $I(Y = y : X)$  can turn out to be negative - which definitely contradicts our naive notion of information content. How is

this possible? The concept of information as used in the theory of communication is a probabilistic notion, which is natural for information transmission over communication channels. Nonetheless, we tend to identify probabilities of messages with frequencies of messages in a sufficiently long sequence, which under some conditions on the stochastic source can be rigorously justified. The great probabilist, Kolmogorov, remarks, ‘If something goes wrong here, the problem lies in the vagueness of our ideas of the relation between mathematical probability theory and real random events in general’. The algorithmic mutual information we introduce below can never be negative, and in this sense is closer to the intuitive notion of information content.

### 2.5.2 Algorithmic mutual information

**Conditional Kolmogorov complexity** To prepare for the definition of Shannon mutual information, we first needed to introduce a conditional version of entropy. Analogously, to prepare for the definition of algorithmic mutual information, we need a notion of conditional Kolmogorov complexity. Intuitively, the conditional prefix Kolmogorov complexity  $K(x|y)$  of  $x$  given  $y$  can be interpreted as the shortest prefix program  $p$  such that, when  $y$  is given to the program  $p$  as input, the program prints  $x$  and then halts. The idea of providing  $p$  with an input  $y$  is realized by putting  $\langle p, y \rangle$  rather than just  $p$  on the input tape of the universal prefix machine  $U$ .

#### Definition 2.4

The conditional prefix Kolmogorov complexity of  $x$  given  $y$  (for free) is

$$K(x|y) = \min_p \{l(p) : U(\langle p, y \rangle) = x, p \in \{0, 1\}^*\}.$$

We define  $K(x) = K(x|\epsilon)$ .

Note that we just redefined  $K(x)$  so that the unconditional Kolmogorov complexity is exactly equal to the conditional Kolmogorov complexity with empty input. This does not contradict our earlier definition: we can choose a reference prefix machine  $U$  such that  $U(\langle p, \epsilon \rangle) = U(p)$ . Then we automatically have  $K(x) = K(x|\epsilon)$ .

Recall from section 2.2 the notation  $\pm, >$ . By definition,  $K(x, y) = K(\langle x, y \rangle)$ . Trivially, the symmetry property holds:  $K(x, y) \pm K(y, x)$ . An interesting property is the ‘additivity of complexity’ property

$$K(x, y) \pm K(x) + K(y|x^*) \pm K(y) + K(x|y^*), \quad (2.15)$$

where  $x^*$  is the first (in standard enumeration order) shortest prefix program that generates  $x$  and then halts. It is easy to see that  $x^*$  has the same information as the pair  $x, K(x)$ : given  $x^*$  we can compute  $x$  and  $l(x^*) = K(x)$ ; given  $x, K(x)$  we can run all programs simultaneously in dovetailed fashion and select the first program of length  $K(x)$  that halts with output  $x$  as  $x^*$ . (Dovetailed fashion means that in phase  $k$  of the process we run all programs  $i$  for  $j$  steps such that  $i + j = k, k = 1, 2, \dots$ ). Equation (2.15) is the Kolmogorov complexity equivalent of the entropy equality (2.12). That this latter equality holds is true by simply rewriting both sides of the equation according to the definitions of averages of joint and marginal probabilities. In fact, potential

individual differences are averaged out. But in the Kolmogorov complexity case we do nothing like that: it is truly remarkable that additivity of algorithmic information holds for individual objects.

The result (2.15) is due to Gács [36], can be found as theorem 3.9.1 in [71] and has a difficult proof. It is perhaps instructive to point out that the version with just  $x$  and  $y$  in the conditionals doesn't hold with  $\pm$ , but holds up to additive logarithmic terms that cannot be eliminated.

To define the algorithmic mutual information between two individual objects  $x$  and  $y$  with no probabilities involved, it is instructive to first recall the probabilistic notion (2.14). Rewriting (2.14) as

$$I(X, Y) = \sum_x \sum_y p(x, y) (-\log p(x) - \log p(y) + \log p(x, y)),$$

and noting that  $-\log p(s)$  is very close to the length of the prefix-free Shannon-Fano code for  $s$ , we are led to the following definition. The information in  $y$  about  $x$  is defined as

$$I(y : x) = K(x) - K(x|y^*) \stackrel{\pm}{=} K(x) + K(y) - K(x, y), \quad (2.16)$$

where the second equality is a consequence of (2.15) and states that this information is symmetrical,  $I(x : y) \stackrel{\pm}{=} I(y : x)$ , and therefore we can talk about mutual information\*. Theorem 2.4 gave the relationship between entropy and ordinary Kolmogorov complexity; it showed that the entropy of distribution  $P$  is approximately equal to the expected (under  $P$ ) Kolmogorov complexity. Theorem 2.5 gives the analogous result for the mutual information (to facilitate comparison to theorem 2.4, note that  $x$  and  $y$  in (2.17) below may stand for strings of arbitrary length  $n$ ).

### Theorem 2.5

Given a recursive probability mass distribution  $p(x, y)$  over  $(x, y)$  we have

$$I(X; Y) - K(p) \stackrel{+}{\leq} \sum_x \sum_y p(x, y) I(x : y) \stackrel{+}{\leq} I(X; Y) + 2K(p), \quad (2.17)$$

with the additive constant that depending only on  $p$  (it is the length of the shortest prefix-free program that computes  $p(x, y)$  from input  $(x, y)$ ).

Thus, we see that the expectation of the algorithmic mutual information  $I(x : y)$  is close to the probabilistic mutual information  $I(X; Y)$ .

**Interpretation in terms of sequential questions** The algorithmic mutual information  $I(y : x) = K(x) - K(x|y^*)$  which equals  $K(x) - K(x|y)$  up to an additive logarithmic term  $O(\log K(y))$  is the savings in number of questions B needs to ask to get to know  $x$  if B already knows  $y$ . Clearly, if  $y$  is the empty word, no information at all, then B needs to ask  $K(x)$  yes-no questions to obtain the consecutive bits of  $x^*$ . But if B

\* The notation of the algorithmic (individual) notion  $I(x : y)$  distinguishes it from the probabilistic (average) notion  $I(X; Y)$ . We deviate slightly from [71] where  $I(y : x)$  is defined as  $K(x) - K(x|y)$ .

already knows  $y$  then he needs to ask only  $K(x|y)$  such questions to obtain the shortest program to compute from  $y$  to  $x$ . The caveat being, as usual, that B has arbitrary amounts of time and storage to perform its computation from  $x$  to  $y$ . For specific individual  $x, y$  this number can be far less than the average as given by Shannon's mutual information.

**Problem and lacuna** Entropy, Kolmogorov complexity and mutual (algorithmic) information are concepts that do not distinguish between different kinds of information (such as 'meaningful' and 'meaningless' information). Such more refined notions can be arrived at by constraining the description methods with which strings are allowed to be encoded, and by considering lossy rather than lossless encoding. Yet the basic notions entropy, Kolmogorov complexity and mutual information continue to play a fundamental rôle. The two most important developments are rate-distortion theory in the Shannon setting ([102], [20]), dealing with 'useful' information, and the Kolmogorov structure function in Kolmogorov's setting, dealing with 'meaningful' information ([66], [103], [20], [37], [116], [119], [98]). It is here that the two theories may have something relevant to say about the notions of 'information' that are studied within the logic and semantics of natural language communities [114]. We briefly illustrate this for the rate-distortion theory.

## 2.6 Shannon's rate distortion: information in questions

As before, we consider a situation in which sender A wants to communicate the outcome of random variable  $X$  to receiver B. The distribution of  $X$  is known to both A and B. But now A is only allowed to use a finite number, say  $R$  bits, to communicate, so that A can only send  $2^R$  different messages. Then the encoding function  $E$  has to map  $\mathcal{X}$  to  $\{0, 1\}^R$ , and  $D$  has to map  $\{0, 1\}^R$  back to  $\mathcal{X}$ . If  $|\mathcal{X}| > 2^R$  or if  $\mathcal{X}$  is uncountable (say,  $\mathcal{X} = \mathbb{R}$ ), then there can be no code  $(D, E)$  such that for all  $x$ ,  $D(E(x)) = x$ . Thus, A and B cannot make sure that  $x$  can always be reconstructed. As the next best thing, they may agree on a code such that for all  $x$ ,  $D(E(x))$  is in some sense 'as close as possible' to the original  $x$ . To formalize this for a given code  $(D, E)$ , we define  $\hat{X} : \mathcal{X} \rightarrow \mathcal{X}$  as the function  $\hat{X}(x) := D(E(x))$ , and we let  $\hat{\mathcal{X}}$  be the range of  $\hat{X}$ . We may interpret  $\hat{X}(x)$  as an estimate of  $x$ , and  $\hat{\mathcal{X}}$  as the set of values it can take. We assume that the 'goodness' of  $\hat{X}(x)$  as an approximation of  $x$  is measured using some distortion function  $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ . This distortion function may be anything that is appropriate to the situation at hand. Once  $d$  is fixed, we may consider the expected distortion

$$E(d(X, \hat{X})) = \sum_{x \in \mathcal{X}} p_x d(x, \hat{X}(x)), \quad (2.18)$$

where, if  $\mathcal{X} = \mathbb{R}$ , the sum is replaced by an integral and  $p_x$  stands for the probability density of  $x$  with respect to Lebesgue measure.

In the rate distortion setting, the goal of A and B is to determine the code  $(D, E)$  with associated  $\hat{X}$  that minimizes the expected distortion.

### Example 2.6

Suppose  $X$  is a real-valued, normally (Gaussian) distributed random variable with

mean  $E(X) = 0$  and variance  $E(X - E(X))^2 = \sigma^2$ . Let us use the squared Euclidean distance  $d(x, \hat{x}) = (x - \hat{x})^2$  as a distortion measure. If A is allowed to use  $R$  bits, then  $\hat{\mathcal{X}}$  can have no more than  $2^R$  elements, whereas  $\mathcal{X} = \mathbb{R}$  is uncountably infinite. We should choose  $\hat{\mathcal{X}}$  and the function  $\hat{X}$  such that (2.18) is minimized. Suppose first  $R = 1$ . Then the optimal  $\hat{X}$  turns out to be

$$\hat{X}(x) = \begin{cases} \sqrt{\frac{2}{\pi}}\sigma^2 & \text{if } x \geq 0 \\ -\sqrt{\frac{2}{\pi}}\sigma^2 & \text{if } x < 0. \end{cases}$$

Thus, the domain  $\mathcal{X}$  is partitioned into two regions, one corresponding to  $x \geq 0$ , and one to  $x < 0$ . That the boundary should be at  $x = 0$  is evident by the symmetry of the Gaussian distribution around 0. Within each region, one then picks a ‘representative point’ so as to minimize (2.18). Similarly, if  $R = 2$ , then  $\mathcal{X}$  should be partitioned into 4 regions, each of which are to be represented by a single point such that (2.18) is minimized. An extreme case is  $R = 0$ : how should B estimate  $X$  if it is not given any information whatsoever? This means that  $\hat{X}(x)$  must take the same value for all  $x$ . The expected distortion (2.18) is then minimized if B picks  $\hat{X} \equiv 0$ , giving distortion equal to  $\sigma^2$ .  $\diamond$

There is no reason in general that the distortion function should be symmetric: in fact, it may be anything that pertains to the situation at hand. It can be considered as (minus) a utility function, indicating the loss that B incurs if he has to predict  $x$  without knowing its precise value.

**Interpretation in terms of sequential questions** Previously, we interpreted entropy as the expected minimum number of yes/no-questions that receiver has to ask to sender in order to determine the precise outcome  $x$  of a random variable  $X$ .

The present setting can be interpreted in terms of a more involved question-and-answer game: now receiver is allowed to ask only  $R$  yes/no-questions. He then has to come up with a guess  $\hat{x}$  of the outcome  $x$ . The quality of this guess is measured by  $d(x, \hat{x})$ . The goal of the receiver is now to ask the  $R$  ‘cleverest possible questions’ that reduce his expected distortion as much as possible; equivalently, they increase his expected utility as much as possible. Thus, there is a relation to ‘quality and quantity of information exchange’ [114] as studied in natural language semantics.

As a concrete case, if  $R = 1$ , then in the Gaussian example above, receiver should ask ‘Is  $x \in [0, \infty)$  or not?’. Every other question reduces the expected distortion by a lesser amount. In general, the present question-and-answer game is very different from the original game where the goal was to minimize the total number of questions. But the following example shows that, if we take a special distortion measure, then the goal of minimizing distortion and minimizing total number of questions are reconciled.

### Example 2.7

Suppose receiver wants to estimate the actual  $x$  by a probability distribution  $P$  on  $\mathcal{X}$ . Thus, if  $R$  bits are allowed to be used, one of  $R$  different distributions on  $\mathcal{X}$  can be sent to receiver. The best that can be done is to partition  $\mathcal{X}$  into  $2^R$  subsets  $\mathcal{A}_1, \dots, \mathcal{A}_{2^R}$ . Sender observes the  $i$  such that  $x \in \mathcal{A}_i$  and passes this information on to receiver. A little thought reveals that the information  $i$  tells the receiver that  $X$  is now distributed

according to the conditional distribution  $P(X = \cdot | X \in \mathcal{A}_i)$ . It is then natural to measure the quality of distribution  $P(X = \cdot | X \in \mathcal{A}_i)$  by its entropy, i.e. by the additional number of questions that receiver has to ask before he knows the value of  $x$  with certainty. That is, we take  $d(x, P) = -\log P(x)$ : the distortion function is the Shannon-Fano code length for the communicated distribution. Here we implicitly generalized the definition of ‘distortion’ measure: we do not require the estimates  $\hat{X}$  to take values in  $\mathcal{X}$  any more. Rather, they are now a set of probability distributions on  $\mathcal{X}$ ; the new definition includes the former as a special case.

With  $d(x, P) = -\log P(x)$ , the expected distortion is  $E(d(X, P)) = H(P)$ . The minimum achievable distortion  $d^*(r)$  for  $R = r$  is given by

$$d^*(r) = \min I(Y; X),$$

where  $\mathcal{Y} = \{1, \dots, 2^R\}$ , and the minimum is over all sets  $\mathcal{Y}$  and all distributions  $P^*$  over  $\mathcal{X} \times \mathcal{Y}$  such that for all  $y \in \mathcal{Y}$ ,  $P^*(Y = y) = \sum_x P^*(Y = y, X = x) = P(Y = y)$ . In particular, for  $r = 0$ ,  $d^*(r) = H(P)$ ; for  $r \geq H(P)$ ,  $d^*(r) = 0$ ; for general  $r$ ,  $d^*(r)$  is the minimum expected number of questions that B still has to ask to determine  $x$ , just after B has already been given the answers to the first  $r$  questions.

Thus, if we pick the Shannon-Fano code length as the distortion measure, then the rate-distortion theory is reconciled with the lossless compression theory. In this case, the distortion-rate function  $d^*(r)$  shows how fast the entropy decreases (the information gained by receiver increases) if receiver always asks the ‘cleverest possible question’, that has the highest expected information gain.  $\diamond$

**Rate distortion and mutual information** As  $R$  increases, the minimum achievable distortion becomes smaller and smaller. Shannon was interested in studying the functional relationship between  $R$  and the minimum achievable distortion  $d^*$  for a given  $R$ . This is called the distortion-rate function. For technical reasons it is often more convenient to study  $R$  as a function of  $d^*$ . This is the celebrated rate-distortion function. As one of the main results in his original paper, Shannon [102] showed that there is a deep connection between the mutual information and the rate-distortion function which holds no matter what distortion function  $d$  is used - thus not only for the Shannon-Fano distortion. We only mention this result because it illustrates that mutual information is a fundamental notion; for a precise statement we refer to [20].

## 2.7 Discussion

Of the three most important developments in Shannon’s original paper, we only discussed two: the noiseless coding theorem for lossless compression (theorem 2.3) and the notion of rate-distortion related to lossy compression. We did not discuss the channel coding theorem, which is related to lossless communication over a noisy channel. These and many other topics in Shannon information theory are thoroughly discussed and explained in the standard reference [20].

Kolmogorov complexity has many applications which we could not discuss here. It leads to a formal notion of randomness of individual sequences that does not refer to

an underlying probability distribution. Also, it lies at the basis of a powerful mathematical theory of inductive inference. Third, it has led to a new mathematical proof technique called the incompressibility method. These and many other topics in Kolmogorov complexity are thoroughly discussed and explained in the standard reference [71]. We end by mentioning a recent development: the Kolmogorov structure function.

The Kolmogorov structure function ([66], [103], [20], [37], [116], [119], [98]) can be viewed (to some extent) as the analogue in Kolmogorov's theory of Shannon's rate distortion. It is based on encoding objects (strings) in two parts: a structural and a random part. We encountered a very simple example of such a description in example 2.3, where we first encoded the frequency of ones in a string (a very simple 'structure') and then the particular sequence with the given frequency (corresponding to the 'random' part of the description). Intuitively, the 'meaning' of the string resides in the structural part and the size of the structural part quantifies the 'meaningful' information in the message. Recently, there have been many new results in this area [116]. Kolmogorov's structure function is closely related to J. Rissanen's minimum description length (MDL) principle for inductive inference. In its simplest guise, this says that the best theory for a given set of data is the theory that minimizes the description length of the theory plus the description length of the data given the theory. Thus, data is encoded by first encoding a theory (constituting the 'structural' part of the data) and then encoding the data using the properties of the data that are prescribed by the theory. Picking the theory minimizing the total description length leads to an automatic trade-off between complexity of the chosen theory and its goodness of fit on the data. This provides a practical and successful principle of inductive inference that may be viewed as a mathematical formalization of 'Occam's razor'\*. But that is quite another story - we refer to [47] and [93] for details. In the next chapter we give an introduction to the minimum description length principle in an entirely non-technical way.

\* In short, this means that 'the simplest explanation is the best'.





### 3. Introducing MDL

#### 3.1 Introduction and overview

How does one decide among competing explanations of data given limited observations? This is the problem of model selection. It stands out as one of the most important problems of inductive and statistical inference. The minimum description length (MDL) principle is a relatively recent method for inductive inference that provides a generic solution to the model selection problem. MDL is based on the following insight: any regularity in the data can be used to compress the data, i.e. to describe it using fewer symbols than the number of symbols needed to describe the data literally. The more regularities there are, the more the data can be compressed. Equating 'learning' with 'finding regularity', we can therefore say that the more we are able to compress the data, the more we have learned about the data. Formalizing this idea leads to a general theory of inductive inference with several attractive properties:

1. **Occam's razor\*** MDL chooses a model that trades-off goodness-of-fit on the observed data with 'complexity' or 'richness' of the model. As such, MDL embodies a form of Occam's razor, a principle that is both intuitively appealing and informally applied throughout all the sciences.
2. **No overfitting, automatically** MDL procedures automatically and inherently protection against overfitting and can be used to estimate both the parameters and the structure (e.g., number of parameters) of a model. In contrast, to avoid overfitting when estimating the structure of a model, traditional methods such as maximum likelihood must be modified and extended with additional, typically ad hoc principles.
3. **Bayesian interpretation** MDL is closely related to Bayesian inference, but avoids some of the interpretation difficulties of the Bayesian approach<sup>†</sup>, especially in the realistic case when it is known *a priori* to the modeler that none of the models under consideration is completely true. In fact:
4. **No need for 'underlying truth'** In contrast to other statistical methods, MDL procedures have a clear interpretation independent of the existence of some underlying 'true' model.
5. **Predictive interpretation** Because data compression is formally equivalent to a form of probabilistic prediction, MDL methods can be interpreted as searching for a model with good predictive performance on unseen data.

In this chapter, we introduce the MDL principle in an entirely non-technical way, concentrating on its most important applications, model selection and avoidance of overfitting. In section 3.2 we discuss the relation between learning and data compression. Section 3.3 introduces model selection and outlines a first, 'crude' version of MDL that can be applied to model selection. Section 3.4 indicates how these 'crude' ideas need to be refined to tackle small sample sizes and differences in model complexity

\* Occam's razor principle: 'Entities should not be multiplied beyond necessity.' Arguably, this serves as a reason to refute David Bohm's [11] [12] alternative deterministic quantum theory, where additional entities like 'hidden' variables and a 'quantum potential' are introduced.

<sup>†</sup> See section 4.8.2, example 4.20.

between models with the same number of parameters. Section 3.5 discusses the philosophy underlying MDL, and section 3.6 considers its relation to Occam's razor. Section 3.7 briefly discusses the history of MDL. All this is summarized in section 3.8.

### 3.2 The fundamental idea: learning as data compression

We are interested in developing a method for learning the laws and regularities in data. The following example will illustrate what we mean by this and give a first idea of how it can be related to descriptions of data.

**Regularity ...** Consider the following three sequences. We assume that each sequence is 10000 bits long, and we just list the beginning and the end of each sequence:

00010001000100010001 ... 000100010001000100010001, (3.1)

01110100110100100110 ... 1010111010111011000101100010, (3.2)

00011000001010100000 ... 0010001000010000001000110000. (3.3)

The first of these three sequences is a 2500-fold repetition of 0001. Intuitively, the sequence looks regular; there seems to be a simple 'law' underlying it; it might make sense to conjecture that future data will also be subject to this law, and to predict that future data will behave according to this law. The second sequence has been generated by tosses of a fair coin. It is intuitively speaking as 'random as possible', and in this sense there is no regularity underlying it. Indeed, we cannot seem to find such a regularity either when we look at the data. The third sequence contains approximately four times as many 0s as 1s. It looks less regular, more random than the first; but it looks less random than the second. There is still some discernible regularity in these data, but of a statistical rather than of a deterministic kind. Again, noticing that such a regularity is there and predicting that future data will behave according to the same regularity seems sensible.

**... and Compression** We claimed that any regularity detected in the data can be used to compress the data, i.e. to describe it in a short manner. Descriptions are always relative to some description method which maps descriptions  $D'$  in a unique manner to data sets  $D$ . A particularly versatile description method is a general-purpose computer language like C or Pascal. A description of  $D$  is then any computer program that prints  $D$  and then halts. Let us see whether our claim works for the three sequences above. Using a language similar to Pascal, we can write a program

```
for i = 1 to 2500; print '0001'; next; halt,
```

which prints sequence (3.1) but is clearly a lot shorter. Thus, sequence (3.1) is indeed highly compressible. On the other hand, we show in section 4.1, that if one generates a sequence like (3.2) by tosses of a fair coin, then with extremely high probability, the shortest program that prints (3.2) and then halts will look something like this:

```
print '01110100110100100110 ... 1010111010111011000101100010'; halt.
```

This program's size is about equal to the length of the sequence. Clearly, it does nothing more than repeat the sequence.

The third sequence lies in between the first two: generalizing  $n = 10000$  to arbitrary length  $n$ , we show in section 4.1 that the first sequence can be compressed to  $\mathcal{O}(\log n)$  bits; with overwhelming probability, the second sequence cannot be compressed at all; and the third sequence can be compressed to some length  $\alpha n$ , with  $0 < \alpha < 1$ .

### Example 3.1 (Compressing various regular sequences)

The regularities underlying sequences (3.1) and (3.3) were of a very particular kind. To illustrate that any type of regularity in a sequence may be exploited to compress that sequence, we give a few more examples:

**The Number  $\pi$**  Evidently, there exists a computer program for generating the first  $n$  digits of  $\pi$  - such a program could be based, for example, on an infinite series expansion of  $\pi$ . This computer program has constant size, except for the specification of  $n$  which takes no more than  $\mathcal{O}(\log n)$  bits. Thus, when  $n$  is very large, the size of the program generating the first  $n$  digits of  $\pi$  will be very small compared to  $n$ : the  $\pi$ -digit sequence is deterministic, and therefore extremely regular.

**Physics data** Consider a two-column table where the first column contains numbers representing various heights from which an object was dropped. The second column contains the corresponding times it took for the object to reach the ground. Assume both heights and times are recorded to some finite precision. In section 3.3 we illustrate that such a table can be substantially compressed by first describing the coefficients of the second-degree polynomial  $H$  that expresses Newton's law; then describing the heights; and then describing the deviation of the time points from the numbers predicted by  $H$ .

**Natural language** Most sequences of words are not valid sentences according to the English language. This fact can be exploited to substantially compress English text, as long as it is syntactically mostly correct: by first describing a grammar for English, and then describing an English text  $D$  with the help of that grammar [43],  $D$  can be described using much less bits than are needed without the assumption that word order is constrained.  $\diamond$

### 3.2.1 Kolmogorov complexity and ideal MDL

To formalize our ideas, we need to decide on a description method, that is, a formal language in which to express properties of the data. The most general choice is a general-purpose\* computer language such as C or Pascal. This choice leads to the definition of the Kolmogorov complexity [71] of a sequence as the length of the shortest program that prints the sequence and then halts. The lower the Kolmogorov complexity of a sequence, the more regular it is. This notion seems to be highly dependent on the particular computer language used. However, it turns out that for every two general-purpose programming languages A and B and every data sequence  $D$ , the length of the shortest program for  $D$  written in language A and the length of the shortest program for  $D$  written in language B differ by no more than a constant  $c$ , which does not depend on the length of  $D$ . This so-called invariance theorem says that, as long as the sequence  $D$  is long enough, it is not essential which computer language one chooses, as long as it is general-purpose. Kolmogorov complexity was introduced, and the invariance theorem

\* By this we mean that a universal Turing machine can be implemented in it [71].

was proved, independently in [65], [16] and [106]. Solomonoff's paper, called 'A theory of inductive inference', contained the idea that the ultimate model for a sequence of data may be identified with the shortest program that prints the data. Solomonoff's ideas were later extended by several authors, leading to an 'idealized' version of MDL ([107], [71], [37]). This idealized MDL is very general in scope, but not practically applicable, for the following two reasons:

1. **Uncomputability** It can be shown<sup>†</sup> that there exists no computer program that, for every set of data  $D$ , when given  $D$  as input, returns the shortest program that prints  $D$  [71].
2. **Arbitrariness/dependence on syntax** In practice we are confronted with small data samples for which the invariance theorem does not say much. Then the hypothesis chosen by idealized MDL may depend on arbitrary details of the syntax of the programming language under consideration.

### 3.2.2 Practical MDL

Like most authors in the field, we concentrate here on non-idealized, practical versions of MDL that explicitly deal with the two problems mentioned above. The basic idea is to scale down Solomonoff's approach so that it does become applicable. This is achieved by using description methods that are less expressive than general-purpose computer languages. Such description methods  $C$  should be restrictive enough so that for any data sequence  $D$ , we can always compute the length of the shortest description of  $D$  that is attainable using method  $C$ ; but they should be general enough to allow us to compress many of the intuitively 'regular' sequences. The price we pay is that, using the 'practical' MDL principle, there will always be some regular sequences which we will not be able to compress. But we already know that there can be no method for inductive inference at all which will always give us all the regularity there is - simply because there can be no automated method which for any sequence  $D$  finds the shortest computer program that prints  $D$  and then halts. Moreover, it will often be possible to guide a suitable choice of  $C$  by *a priori* knowledge we have about our problem domain. For example, below we consider a description method  $C$  that is based on the class of all polynomials, such that with the help of  $C$  we can compress all data sets which can meaningfully be seen as points on some polynomial.

## 3.3 MDL and model selection

Let us recapitulate our main insights so far:

### MDL: The basic idea

The goal of statistical inference may be cast as trying to find regularity in the data. 'Regularity' may be identified with 'ability to compress'. MDL combines these two insights by viewing learning as data compression: it tells us that, for a given set of hypotheses  $\mathcal{H}$  and data set  $D$ , we should try to find the hypothesis or combination of hypotheses in  $\mathcal{H}$  that compresses  $D$  most.

<sup>†</sup> This follows from Gödel's incompleteness theorem, for a popular discussion see [56].

This idea can be applied to all sorts of inductive inference problems, but it turns out to be most fruitful in (and its development has mostly concentrated on) problems of model selection and, more generally, dealing with overfitting. Here is a standard example (we explain the difference between ‘model’ and ‘hypothesis’ after the example).

### Example 3.2 (Model selection and overfitting)

Consider the points in figure 3.1. We would like to learn how the  $y$ -values depend on the  $x$ -values. To this end, we may want to fit a polynomial to the points. Straightforward linear regression will give us the leftmost polynomial - a straight line that seems overly simple: it does not capture the regularities in the data well. Since for any set of  $n$  points there exists a polynomial of the  $(n - 1)$ -st degree that goes exactly through all these points, simply looking for the polynomial with the least error will give us a polynomial like the one in the second picture. This polynomial seems overly complex: it reflects the random fluctuations in the data rather than the general pattern underlying it. Instead of picking the overly simple or the overly complex polynomial, it seems more reasonable to prefer a relatively simple polynomial with small but nonzero error, as in the rightmost picture.

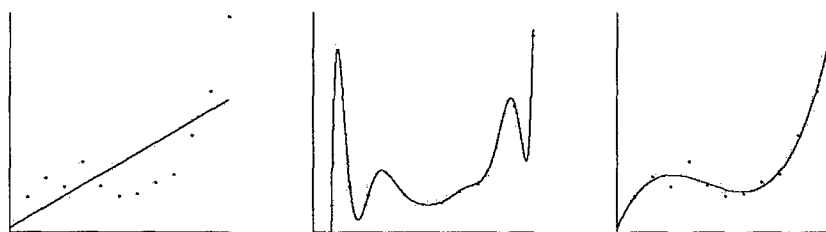


Figure 3.1: A simple, a complex and a trade-off (3rd degree) polynomial.

This intuition is confirmed by numerous experiments on real-world data from a broad variety of sources [93], [115], [87]: if one naively fits a high-degree polynomial to a small sample (set of data points), then one obtains a very good fit to the data. Yet if one tests the inferred polynomial on a second set of data coming from the same source, it typically fits this test data very badly in the sense that there is a large distance between the polynomial and the new data points. We say that the polynomial overfits the data. Indeed, all model selection methods that are used in practice either implicitly or explicitly choose a trade-off between goodness-of-fit and complexity of the models involved. In practice, such trade-offs lead to much better predictions of test data than one would get by adopting the ‘simplest’ (one degree) or most ‘complex’ ( $n - 1$ -degree) polynomial. MDL provides a means of achieving such a trade-off.  $\diamond$

It will be useful to make a precise distinction between ‘model’ and ‘hypothesis’:

\* Strictly speaking, in our context it is not very accurate to speak of ‘simple’ or ‘complex’ polynomials; instead we should call the set of first degree polynomials ‘simple’, and the set of 100-th degree polynomials ‘complex’.

**Models vs. Hypotheses**

We use the phrase point hypothesis to refer to a single probability distribution or function. An example is the polynomial  $5x^2 + 4x + 3$ . A point hypothesis is also known as a ‘simple hypothesis’ in the statistical literature. We use the word model to refer to a family (set) of probability distributions or functions with the same functional form. An example is the set of all second-degree polynomials. A model is also known as a ‘composite hypothesis’ in the statistical literature. We use hypothesis as a generic term, referring to both point hypotheses and models.

In our terminology, the problem described in example 3.2 is a ‘hypothesis selection problem’ if we are interested in selecting both the degree of a polynomial and the corresponding parameters; it is a ‘model selection problem’ if we are mainly interested in selecting the degree.

To apply MDL to polynomial or other types of hypothesis and model selection, we have to make precise the somewhat vague insight ‘learning may be viewed as data compression’. This can be done in various ways. In this section, we concentrate on the earliest and simplest implementation of the idea. This is the so-called crude<sup>†</sup> two-part code version of MDL:

**Crude, two-part version of MDL principle (informally stated)**

Let  $\mathcal{H}^{(1)}, \mathcal{H}^{(2)}, \dots$  be a list of candidate models (e.g.,  $\mathcal{H}^{(k)}$  is the set of  $k$ -th degree polynomials), each containing a set of point hypotheses (e.g., individual polynomials). The best point hypothesis  $H \in \mathcal{H}^{(1)} \cup \mathcal{H}^{(2)} \cup \dots$  to explain the data  $D$  is the one which minimizes the sum  $L(H) + L(D|H)$ , where

- $L(H)$  is the length, in bits, of the description of the hypothesis, and
- $L(D|H)$  is the length, in bits, of the description of the data when encoded with the help of the hypothesis.

The best model to explain  $D$  is the smallest model containing the selected  $H$ .

**Example 3.3 (Polynomials continued)**

In our previous example, the candidate hypotheses were polynomials. We can describe a polynomial by describing its coefficients in a certain precision (number of bits per parameter). Thus, the higher the degree of a polynomial or the precision, the more bits we need to describe it and the more ‘complex’ it becomes. A description of the data ‘with the help of’ a hypothesis means that the better the hypothesis fits the data, the shorter the description will be. A hypothesis that fits the data well gives us a lot of information about the data. Such information can always be used to compress the data (section 4.1). Intuitively, this is because we only have to code the errors the hypothesis makes on the data rather than the full data. In our polynomial example, the better a polynomial  $H$  fits  $D$ , the fewer bits we need to encode the discrepancies between the actual  $y$ -values  $y_i$  and the predicted  $y$ -values  $H(x_i)$ . We can typically find a very complex point hypothesis (large  $L(H)$ ) with a very good fit (small  $L(D|H)$ ). We can also typically find a very simple point hypothesis (small  $L(H)$ ) with a rather bad fit (large

<sup>†</sup> The terminology ‘crude MDL’ is not standard. It is introduced here for pedagogical reasons, to clarify the importance of having a single, unified principle for designing codes. It should be noted that Rissanen’s and Barron’s early theoretical papers on MDL already contain such principles, albeit in a slightly different form than in their recent papers. Early practical applications [83], [43] often do use ad hoc two-part codes which really are ‘crude’ in the sense defined here.

$L(D|H)$ ). The sum of the two description lengths will be minimized at a hypothesis that is quite (but not too) ‘simple’, with a good (but not perfect) fit.  $\diamond$

### 3.4 Crude and refined MDL

Crude MDL picks the  $H$  minimizing the sum  $L(H) + L(D|H)$ . To make this procedure well-defined, we need to agree on precise definitions for the codes (description methods) giving rise to lengths  $L(D|H)$  and  $L(H)$ . We now discuss these codes in more detail. We will see that the definition of  $L(H)$  is problematic, indicating that we somehow need to ‘refine’ our crude MDL principle.

**Definition of  $L(D|H)$**  Consider a two-part code as described above, and assume for the time being that all  $H$  under consideration define probability distributions. If  $H$  is a polynomial, we can turn it into a distribution by making the additional assumption that the  $Y$ -values are given by  $Y = H(X) + Z$ , where  $Z$  is a normally distributed noise term.

For each  $H$  we need to define a code with length  $L(\cdot|H)$  such that  $L(D|H)$  can be interpreted as ‘the code length of  $D$  when encoded with the help of  $H$ ’. It turns out that for probabilistic hypotheses, there is only one reasonable choice for this code. It is the so-called Shannon-Fano code, satisfying, for all data sequences  $D$ ,  $L(D|H) = -\log P(D|H)$ , where  $P(D|H)$  is the probability mass or density of  $D$  according to  $H$  - such a code always exists, section 4.1.

**Definition of  $L(H)$ : a problem for crude MDL** It is more problematic to find a good code for hypotheses  $H$ . Some authors have simply used ‘intuitively reasonable’ codes in the past, but this is not satisfactory: since the description length  $L(H)$  of any fixed point hypothesis  $H$  can be very large under one code, but quite short under another, our procedure is in danger of becoming arbitrary. Instead, we need some additional principle for designing a code for  $H$ . In the first publications on MDL [88], [89], it was advocated to choose some sort of minimax code for  $H$ , minimizing, in some precisely defined sense, the shortest worst-case total description length  $L(H) + L(D|H)$ , where the worst-case is over all possible data sequences. Thus, the MDL principle is employed at a ‘meta-level’ to choose a code for  $H$ . However, this code requires a cumbersome discretization of the model space  $H$ , which is not always feasible in practice. Alternatively, Barron [6] encoded  $H$  by the shortest computer program that, when input  $D$ , computes  $P(D|H)$ . While it can be shown that this leads to similar code lengths, it is computationally problematic. Later, Rissanen [90] realized that these problems could be side-stepped by using a one-part rather than a two-part code. This development culminated in 1996 in a completely precise prescription of MDL for many, but certainly not all practical situations [95]. We call this modern version of MDL ‘refined MDL’.

**Refined MDL** In refined MDL, we associate a code for encoding  $D$  not with a single  $H \in \mathcal{H}$ , but with the full model  $\mathcal{H}$ . Thus, given model  $\mathcal{H}$ , we encode data not in two parts but we design a single one-part code with lengths  $\bar{L}(D|H)$ . This code is designed such that whenever there is a member of (parameter in)  $H$  that fits the data well, in the sense that  $L(D|H)$  is small, then the code length  $\bar{L}(D|H)$  will also be small. Codes with this property are called universal codes in the information-theoretic literature [9].

Among all such universal codes, we pick the one that is minimax optimal in a sense made precise in section 4.4. For example, the set  $H^{(3)}$  of third-degree polynomials is associated with a code with lengths  $\bar{L}(\cdot|H^{(3)})$  such that, the better the data  $D$  are fit by the best-fitting third-degree polynomial, the shorter the code length  $\bar{L}(D|H)$ .  $\bar{L}(D|H)$  is called the stochastic complexity of the data given the model.

**Parametric complexity** The second fundamental concept of refined MDL is the parametric complexity of a parametric model  $H$  which we denote by  $\text{COMP}(\mathcal{H})$ . This is a measure of the ‘richness’ of model  $H$ , indicating its ability to fit random data. This complexity is related to the degrees-of-freedom in  $H$ , but also to the geometrical structure of  $H$ ; see example 3.4. To see how it relates to stochastic complexity, let, for given data  $D$ ,  $\hat{H}$  denote the distribution in  $\mathcal{H}$  which maximizes the concomitant probability, and hence minimizes the code length  $L(D|\hat{H})$  of  $D$ . It turns out that

$$\text{stochastic complexity of } D \text{ given } \mathcal{H} = L(D|\hat{H}) + \text{COMP}(\mathcal{H}).$$

Refined MDL model selection between two parametric models (such as the models of first and second degree polynomials) now proceeds by selecting the model such that the stochastic complexity of the given data  $D$  is smallest. Although we used a one-part code to encode data, refined MDL model selection still involves a trade-off between two terms: a goodness-of-fit term  $L(D|\hat{H})$  and a complexity term  $\text{COMP}(\mathcal{H})$ . However, because we do not explicitly encode hypotheses  $H$  any more, there is no arbitrariness any more. The resulting procedure can be interpreted in several different ways, some of which provide us with rationales for MDL beyond the pure coding interpretation (sections 4.5.1 - 4.5.4):

1. **Counting interpretation** The parametric complexity of a model is the logarithm of the number of essentially different, distinguishable point hypotheses within the model.
2. **Two-part code interpretation** For large samples, the stochastic complexity can be interpreted as a two-part code length of the data after all, where hypotheses  $H$  are encoded with a special code that works by first discretizing the model space  $H$  into a set of ‘maximally distinguishable hypotheses’, and then assigning equal code length to each of these.
3. **Bayesian interpretation** In many cases, refined MDL model selection coincides with Bayes factor model selection based on a non-informative prior such as Jeffreys’ prior [10].
4. **Prequential interpretation** Refined MDL model selection can be interpreted as selecting the model with the best predictive performance when sequentially predicting - prequential - unseen test data, in the sense described in section 4.5.4. This makes it an instance of Dawid’s [22] prequential model validation and also relates it to cross-validation methods.

Refined MDL allows us to compare models of different functional form. It even accounts for the phenomenon that different models with the same number of parameters may not be equally ‘complex’:

#### Example 3.4

Consider two models from psychophysics describing the relationship between physical quantities (e.g., light intensity) and their psychological counterparts (e.g. brightness)



[80]:  $y = ax^b + Z$  (Stevens' model) and  $y = a \ln(x + b) + Z$  (Fechner's model) where  $Z$  is a normally distributed noise term. Both models have two free parameters; nevertheless, it turns out that in a sense, Stevens' model is more flexible or complex than Fechner's. Roughly speaking, this means there are a lot more data patterns that can be explained by Stevens' model than can be explained by Fechner's model. In [80] many samples has been generated of size 4 from Fechner's model, using some fixed parameter values. They then fitted both models to each sample. In 67% of the trials, Stevens' model fitted the data better than Fechner's, even though the latter generated the data. Indeed, in refined MDL, the 'complexity' associated with Stevens' model is much larger than the complexity associated with Fechner's. If both models fit the data equally well, MDL will prefer Fechner's model.  $\diamond$

**Summary** Refined MDL removes the arbitrary aspect of crude, two-part code MDL and associates parametric models with an inherent 'complexity' that does not depend on any particular description method for hypotheses. We should, however, warn the reader that we only discussed a special, simple situation in which we compared a finite number of parametric models that satisfy certain regularity conditions. Whenever the models do not satisfy these conditions, or if we compare an infinite number of models, then the refined ideas have to be extended. We then obtain a 'general' refined MDL principle, which employs a combination of one-part and two-part codes.

### 3.5 The MDL philosophy

The first central MDL idea is that every regularity in data may be used to compress that data; the second central idea is that learning can be equated with finding regularities in data. Whereas the first part is relatively straightforward, the second part of the idea implies that methods for learning from data must have a clear interpretation independent of whether any of the models under consideration is 'true' or not. Quoting J. Rissanen [93], the main originator of MDL:

"We never want to make the false assumption that the observed data actually were generated by a distribution of some kind, say Gaussian, and then go on to analyze the consequences and make further deductions. Our deductions may be entertaining but quite irrelevant to the task at hand, namely, to learn useful properties from the data."

Based on such ideas, Rissanen developed a radical philosophy of learning and statistical inference that is considerably different from the ideas underlying mainstream statistics, both frequentist and Bayesian. We now describe this philosophy in more detail:

**1. Regularity as compression** According to Rissanen, the goal of inductive inference should be to 'squeeze out as much regularity as possible' from the given data. The main task for statistical inference is to distill the meaningful information present in the data, i.e., to separate structure (interpreted as the regularity, the 'meaningful information') from noise (interpreted as the 'accidental information'). For the three sequences of example 3.2, this would amount to the following: the first sequence would be considered as entirely regular and 'noiseless'. The second sequence would be considered

as entirely random - all information in the sequence is accidental, there is no structure present. In the third sequence, the structural part would (roughly) be the pattern that 4 times as many 0s than 1s occur; given this regularity, the description of exactly which of all sequences with four times as many 0s than 1s occurs, is the accidental information.

**2. Models as languages** Rissanen interprets models (sets of hypotheses) as nothing more than languages for describing useful properties of the data - a model  $\mathcal{H}$  is identified with its corresponding universal code  $\bar{L}(\cdot|\mathcal{H})$ . Different individual hypotheses within the models express different regularities in the data, and may simply be regarded as statistics, that is, summaries of certain regularities in the data. These regularities are present and meaningful independently of whether some  $H^* \in \mathcal{H}$  is the 'true state of nature' or not. Suppose that the model  $H$  under consideration is probabilistic. In traditional theories, one typically assumes that some  $P^* \in \mathcal{H}$  generates the data, and then 'noise' is defined as a random quantity relative to this  $P^*$ . In the MDL view 'noise' is defined relative to the model  $\mathcal{H}$  as the residual number of bits needed to encode the data once the model  $\mathcal{H}$  is given. Thus, noise is not a random variable: it is a function only of the chosen model and the actually observed data. Indeed, there is no place for a 'true distribution' or a 'true state of nature' in this view - there are only models and data. To bring out the difference to the ordinary statistical viewpoint, consider the phrase 'these experimental data are quite noisy'. According to a traditional interpretation, such a statement means that the data were generated by a distribution with high variance. According to the MDL philosophy, such a phrase means only that the data are not compressible with the currently hypothesized model - as a matter of principle, it can never be ruled out that there exists a different model under which the data are very compressible (not noisy) after all!

**3. We have only the data** Many (but not all\*) other methods of inductive inference are based on the idea that there exists some 'true state of nature', typically a distribution assumed to lie in some model  $\mathcal{H}$ . The methods are then designed as a means to identify or approximate this state of nature based on as little data as possible. According to Rissanen, such methods are fundamentally flawed. The main reason is that the methods are designed under the assumption that the true state of nature is in the assumed model  $\mathcal{H}$ , which is often not the case. Therefore, such methods only admit a clear interpretation under assumptions that are typically violated in practice. Many cherished statistical methods are designed in this way - we mention hypothesis testing, minimum-variance unbiased estimation, several non-parametric methods, and even some forms of Bayesian inference - see example 4.20. In contrast, MDL has a clear interpretation which depends only on the data, and not on the assumption of any underlying 'state of nature'.

### Example 3.5 (Models that are wrong, yet useful)

Even though the models under consideration are often wrong, they can nevertheless be very useful. Examples are the successful 'Naive Bayes' model for spam filtering, Hidden Markov Models for speech recognition (is speech a stationary ergodic process? probably not), and the use of linear models in econometrics and psychology. Since

\* For example, cross-validation cannot easily be interpreted in such terms of 'a method hunting for the true distribution'.

these models are evidently wrong, it seems strange to base inferences on them using methods that are designed under the assumption that they contain the true distribution. To be fair, we should add that domains such as spam filtering and speech recognition are not what the fathers of modern statistics had in mind when they designed their procedures - they were usually thinking about much simpler domains, where the assumption that some distribution  $P^* \in \mathcal{H}$  is 'true' may not be so unreasonable.  $\diamond$

**4. MDL and consistency** Let  $\mathcal{H}$  be a probabilistic model, such that each  $P \in \mathcal{H}$  is a probability distribution. Roughly, a statistical procedure is called consistent relative to  $\mathcal{H}$  if, for all  $P^* \in \mathcal{H}$ , the following holds. Suppose data are distributed according to  $P^*$ . Then given enough data, the learning method will yield a good approximation of  $P^*$  with high probability. Many traditional statistical methods have been designed with consistency in mind (section 4.2).

The fact that in MDL, we do not assume a true distribution may suggest that we do not care about statistical consistency. But this is not the case: we would still like our statistical method to be such that in the idealized case where one of the distributions in one of the models under consideration actually generates the data, our method is able to identify this distribution, given enough data. If even in the idealized special case where a 'truth' exists within our models, the method fails to learn it, then we certainly cannot trust it to do something reasonable in the more general case, where there may not be a 'true distribution' underlying the data at all. Thus, consistency is important in the MDL philosophy. However, it is used as a sanity check (for a method that has been developed without making distributional assumptions) rather than as a design principle.

In fact, mere consistency is not sufficient. We would like our method to converge to the imagined true  $P^*$  fast, based on as small a sample as possible. Two-part code MDL with 'clever' codes achieves good rates of convergence in this sense (Barron and Cover [8], complemented by [127], show that in many situations, the rates are minimax optimal). The same seems to be true for refined one-part code MDL [9], although there is at least one surprising exception where inference based on the normalized maximum likelihood and Bayesian universal model behaves abnormally - see [21] for the details.

**Summary** The MDL philosophy is quite agnostic about whether any of the models under consideration is 'true', or whether something like a 'true distribution' even exists. Nevertheless, it has been suggested [124], [27] that MDL embodies a naive belief that 'simple models' are '*a priori* more likely to be true' than complex models. Below we explain why such claims are mistaken.

### 3.6 MDL and Occam's razor

When two models fit the data equally well, MDL will choose the one that is the 'simplest' in the sense that it allows for a shorter description of the data. As such, it implements a precise form of Occam's razor - even though as more and more data becomes available, the model selected by MDL may become more and more 'complex'! Occam's razor is sometimes criticized for being either (1) arbitrary or (2) false [124], [27]. Do these criticisms apply to MDL as well?

**1. ‘Occam’s razor (and MDL) is arbitrary’** Because ‘description length’ is a syntactic notion it may seem that MDL selects an arbitrary model: different codes would have led to different description lengths, and therefore, to different models. By changing the encoding method, we can make ‘complex’ things ‘simple’ and vice versa. This overlooks the fact we are not allowed to use just any code we like! ‘Refined’ MDL tells us to use a specific code, independent of any specific parameterization of the model, leading to a notion of complexity that can also be interpreted without any reference to ‘description lengths’ (see also section 4.9.1).

**2. ‘Occam’s razor is false’** It is often claimed that Occam’s razor is false - we often try to model real-world situations that are arbitrarily complex, so why should we favor simple models? In the words of G. Webb: ‘What good are simple models of a complex world?’

The short answer is: even if the true data generating machinery is very complex, it may be a good strategy to prefer simple models for small sample sizes. Thus, MDL (and the corresponding form of Occam’s razor) is a strategy for inferring models from data (‘choose simple models at small sample sizes’), not a statement about how the world works (‘simple models are more likely to be true’) - indeed, a strategy cannot be true or false, it is ‘clever’ or ‘stupid’. And the strategy of preferring simpler models is clever even if the data generating process is highly complex, as illustrated by the following example:

**Example 3.6 (‘Infinitely’ complex sources)**

Suppose that data are subject to the law  $Y = g(X) + Z$  where  $g$  is some continuous function and  $Z$  is some noise term with mean 0. If  $g$  is not a polynomial, but  $X$  only takes values in a finite interval, say  $[-1, 1]$ , we may still approximate  $g$  arbitrarily well by taking higher and higher degree polynomials. For example, let  $g(x) = \exp(x)$ . Note that, the exponential function is computed exploiting polynomial approximations. Then, if we use MDL to learn a polynomial for data  $D = ((x_1, y_1), \dots, (x_n, y_n))$ , the degree of the polynomial  $\tilde{f}^{(n)}$  selected by MDL at sample size  $n$  will increase with  $n$ , and with high probability,  $\tilde{f}^{(n)}$  converges to  $g(x) = \exp(x)$  in the sense that  $\max_{x \in [-1, 1]} |\tilde{f}^{(n)}(x) - g(x)| \rightarrow 0$ . Of course, if we had better prior knowledge about the problem we could have tried to learn  $g$  using a model class  $\mathcal{M}$  containing the function  $y = \exp(x)$ . But in general, both our imagination and our computational resources are limited, and we may be forced to use imperfect models.  $\diamond$

If, based on a small sample, we choose the best-fitting polynomial  $\hat{f}$  within the set of all polynomials, then, even though  $\hat{f}$  will fit the data very well, it is likely to be quite unrelated to the ‘true’  $g$ , and  $\hat{f}$  may lead to disastrous predictions of future data. The reason is that, for small samples, the set of all polynomials is very large - in the sense of the number of elements - compared to the set of possible data patterns that we might have observed. Therefore, any particular data pattern can only give us very limited information about which high-degree polynomial best approximates  $g$ . On the other hand, if we choose the best-fitting  $\hat{f}^\circ$  in some much smaller set such as the set of second-degree polynomials, then it is highly probable that the prediction quality (mean squared error) of  $\hat{f}^\circ$  on future data is about the same as its mean squared error on the data we observed: the size (complexity) of the contemplated model is relatively small compared to the set of possible data patterns that we might have observed. Therefore,

the particular pattern that we do observe gives us a lot of information on what second-degree polynomial best approximates  $g$ .

Thus, (a)  $\hat{f}^\circ$  typically leads to better predictions of future data than  $\hat{f}$ ; and (b) unlike  $\hat{f}$ ,  $\hat{f}^\circ$  is reliable in that it gives a correct impression of how good it will predict future data even if the 'true'  $g$  is 'infinitely' complex. This idea does not just appear in MDL, but is also the basis of Vapnik's [115] structural risk minimization approach and many standard statistical methods for non-parametric inference. In such approaches one acknowledges that the data generating machinery can be infinitely complex (e.g., not describable by a finite degree polynomial). Nevertheless, it is still a good strategy to approximate it by simple hypotheses (low-degree polynomials) as long as the sample size is small. Summarizing:

**The inherent difference between under- and overfitting**

If we choose an overly simple model for our data, then the best-fitting point hypothesis within the model is likely to be almost the best predictor, within the simple model, of future data coming from the same source. If we overfit (choose a very complex model) and there is noise in our data, then, even if the complex model contains the 'true' point hypothesis, the best-fitting point hypothesis within the model is likely to lead to very bad predictions of future data coming from the same source.

This statement is very imprecise and is meant more to convey the general idea than to be completely true. As will become clear in section 4.9.1, it becomes provably true if we use MDL's measure of model complexity; we measure prediction quality by logarithmic loss; and we assume that one of the distributions in  $\mathcal{H}$  actually generates the data.

### 3.7 History

The MDL principle has mainly been developed by J. Rissanen in a series of papers starting with [88]. It has its roots in the theory of Kolmogorov or algorithmic complexity [71], developed in the 1960s by Solomonoff [106], Kolmogorov [65] and Chaitin [16]. Among these authors, Solomonoff (a former student of the famous philosopher of science, Rudolf Carnap) was explicitly interested in inductive inference. His 1964 paper contains explicit suggestions on how the underlying ideas could be made practical, thereby foreshadowing some of the later work on two-part MDL. While Rissanen was not aware of Solomonoff's work at the time, Kolmogorov's [65] paper did serve as an inspiration for Rissanen's [88] development of MDL.

Another important inspiration for Rissanen was Akaike's [1] AIC method for model selection, essentially the first model selection method based on information theoretic ideas. Even though Rissanen was inspired by AIC, both the actual method and the underlying philosophy are substantially different from MDL.

MDL is much closer related to the minimum message length principle, developed by Wallace and his co-workers in a series of papers starting with the ground-breaking [121]; other milestones are [122] and [123]. Remarkably, Wallace developed his ideas without being aware of the notion of Kolmogorov complexity. Although Rissanen became aware of Wallace's work before the publication of [88], he developed his ideas

mostly independently, being influenced rather by Akaike and Kolmogorov. Indeed, despite the close resemblance of both methods in practice, the underlying philosophy is quite different - see section 4.8.

The first publications on MDL only mention two-part codes. Important progress was made by Rissanen [90], in which prequential codes are employed for the first time and [92], introducing the Bayesian mixture codes into MDL. This led to the development of the notion of stochastic complexity as the shortest code length of the data given a model [91], [92]. However, the connection to Shtarkov's normalized maximum likelihood code [104] was not made until 1996, and this prevented the full development of the notion of 'parametric complexity'. In the mean time, in his impressive Ph.D. thesis, Barron [6] showed how a specific version of the two-part code criterion has excellent frequentist statistical consistency properties. This was extended by Barron and Cover [8] who achieved a breakthrough for two-part codes: they gave clear prescriptions on how to design codes for hypotheses, relating codes with good minimax code length properties to rates of convergence in statistical consistency theorems. Some of the ideas of Rissanen [92] and Barron and Cover [8] were, as it were, unified when Rissanen [95] introduced a new definition of stochastic complexity based on the normalized maximum likelihood code (section 4.4). The resulting theory was summarized for the first time by Barron, Rissanen, and Yu [9], and is called 'refined MDL' in the present overview.

### 3.8 Summary and outlook

We discussed how regularity is related to data compression, and how MDL employs this connection by viewing learning in terms of data compression. One can make this precise in several ways; in idealized MDL one looks for the shortest program that generates the given data. This approach is not feasible in practice, and here we concern ourselves with practical MDL. Practical MDL comes in a crude version based on two-part codes and in a modern, more refined version based on the concept of universal coding. The basic ideas underlying all these approaches can be found in the boxes spread throughout the text.

These methods are mostly applied to model selection but can also be used for other problems of inductive inference. In contrast to most existing statistical methodology, they can be given a clear interpretation irrespective of whether or not there exists some 'true' distribution generating data - inductive inference is seen as a search for regular properties in (interesting statistics of) the data, and there is no need to assume anything outside the model and the data. In contrast to what is sometimes thought, there is no implicit belief that 'simpler models are more likely to be true' - MDL does embody a preference for 'simple' models, but this is best seen as a strategy for inference that can be useful even if the environment is not simple at all.

In the next chapter, we formally introduce both the crude and the refined versions of practical MDL. For this, it is absolutely essential that the reader familiarizes him- or herself with two basic notions of coding and information theory: the relation between code length functions and probability distributions, and (for refined MDL), the idea of universal coding - a large part of the chapter will be devoted to these.

## 4. Minimum description length

In chapter 3, we introduced the MDL principle in an informal way. In this chapter, we give an introduction to MDL that is mathematically precise. Throughout the text, we assume some basic familiarity with probability theory. While some prior exposure to basic statistics is highly useful, it is not required. The chapter can be read without any prior knowledge of information theory, and is organized as follows:

- The first two sections are of a preliminary nature:
  - Any understanding of MDL requires some minimal knowledge of information theory - in particular the relationship between probability distributions and codes. This relationship is explained in section 4.1.
  - Relevant statistical notions such as ‘maximum likelihood estimation’ are reviewed in section 4.2. There we also introduce the Markov chain model which will serve as an example model throughout the text.
- Based on this preliminary material, we formalize a simple version of the MDL principle in section 4.3. In this text it is called the crude two-part MDL principle. We explain why, for successful practical applications, crude MDL needs to be refined.
- Section 4.4 is once again preliminary: it discusses universal coding, the information theoretic concept underlying refined versions of MDL.
- Sections 4.5 - 4.7 define and discuss refined MDL. They are the key sections of the chapter:
  - Section 4.5 discusses basic refined MDL for comparing a finite number of simple statistical models and introduces the central concepts of parametric and stochastic complexity. It gives an asymptotic expansion of these quantities and interprets them from a compression, a geometric, a Bayesian and a predictive point of view.
  - Section 4.6 extends refined MDL to harder model selection problems, and in doing so reveals the general, unifying idea.
  - Section 4.7 briefly discusses how to extend MDL to applications beyond model selection.
- The next two sections place ‘refined MDL’ in its context:
  - Section 4.8 compares MDL to other approaches to inductive inference, most notably the related but different Bayesian approach.
  - Section 4.9 discusses perceived as well as real problems with MDL. The perceived problems relate to MDL’s relation to Occam’s razor, the real problems relate to the fact that applications of MDL sometimes perform suboptimally in practice.
- Finally, section 4.10 provides a discussion.

Throughout the text, paragraph headings reflect the most important concepts. Boxes summarize the most important findings. Together, paragraph headings and boxes provide an overview of MDL theory.

## 4.1 Information theory I: probabilities and code lengths

This first section is a mini-primer on information theory, focusing on the relationship between probability distributions and codes. A good understanding of this relationship is essential for a good understanding of MDL. After some preliminaries, section 4.1.1 introduces prefix codes, the type of codes we work with in MDL. These are related to probability distributions in two ways. In section 4.1.2 we discuss the first relationship, which is related to the Kraft inequality: for every probability mass function  $P$ , there exists a code with length  $\lceil -\log P \rceil$ , and vice versa. The symbol  $\lceil \cdot \rceil$  is defined below. Section 4.1.3 discusses the second relationship, related to the information inequality, which says that if the data are distributed according to  $P$ , then the code with length  $\lceil -\log P \rceil$  achieves the minimum expected code length. Throughout the section we give examples relating our findings to our discussion of regularity and compression in section 3.2 of chapter 3.

**Preliminaries and notational conventions - codes** We use  $\log$  to denote logarithm to base 2. For real-valued  $x$  we use  $\lceil x \rceil$  to denote the ceiling of  $x$ , that is,  $x$  rounded up to the nearest integer. We often abbreviate  $x_1, \dots, x_n$  to  $x_n$ . Let  $\mathcal{X}$  be a finite or countable set. A code for  $\mathcal{X}$  is defined as a 1-to-1 mapping from  $\mathcal{X}$  to  $\cup_{n \geq 1} \{0, 1\}^n$ .  $\cup_{n \geq 1} \{0, 1\}^n$  is the set of binary strings (sequences of 0s and 1s) of length 1 or larger. For a given code  $C$ , we use  $C(x)$  to denote the encoding of  $x$ . Every code  $C$  induces a function  $L_C : \mathcal{X} \rightarrow \mathbb{N}$  called the code length function.  $L_C(x)$  is the number of bits (symbols) needed to encode  $x$  using code  $C$ .

Our definition of code implies that we only consider lossless encoding in MDL\*: for any description  $z$  it is always possible to retrieve the unique  $x$  that gave rise to it. More precisely, because the code  $C$  must be 1-to-1, there is at most one  $x$  with  $C(x) = z$ . Then  $x = C^{-1}(z)$ , where the inverse  $C^{-1}$  of  $C$  is sometimes called a ‘decoding function’ or ‘description method’.

**Preliminaries and notational conventions - probability** Let  $P$  be a probability distribution defined on a finite or countable set  $\mathcal{X}$ . We use  $P(x)$  to denote the probability of  $x$ , and we denote the corresponding random variable by  $X$ . If  $P$  is a function on finite or countable  $\mathcal{X}$  such that  $\sum_x P(x) < 1$ , we call  $P$  a defective distribution. A defective distribution may be thought of as a probability distribution that puts some of its mass on an imagined outcome that in reality will never appear.

A probabilistic source  $P$  is a sequence of probability distributions  $P^{(1)}, P^{(2)}, \dots$  on  $\mathcal{X}^1, \mathcal{X}^2, \dots$  such that for all  $n$ ,  $P^{(n)}$  and  $P^{(n+1)}$  are compatible:  $P^{(n)}$  is equal to the ‘marginal’ distribution of  $P^{(n+1)}$  restricted to  $n$  outcomes. That is, for all  $x^n \in \mathcal{X}^n$ ,  $P^{(n)}(x^n) = \sum_{y \in \mathcal{X}} P^{(n+1)}(x^n, y)$ . Whenever this cannot cause any confusion, we write  $P(x^n)$  rather than  $P^{(n)}(x^n)$ . A probabilistic source may be thought of as a probability distribution on infinite sequences<sup>†</sup>. We say that the data are i.i.d. (independently and identically distributed) under source  $P$  if for each  $n$ ,  $x^n \in \mathcal{X}^n$ ,  $P(x^n) = \prod_{i=1}^n P(x_i)$ .

\* However, see section 4.8.4.

† Working directly with distributions on infinite sequences is more elegant, but it requires measure theory, which we want to avoid here.



#### 4.1.1 Prefix codes

In MDL we only work with a subset of all possible codes, the so-called prefix codes. A prefix code<sup>‡</sup> is a code such that no code word is a prefix of any other code word. For example, let  $X = \{a, b, c\}$ . Then the code  $C_1$  defined by  $C_1(a) = 0$ ,  $C_1(b) = 10$ ,  $C_1(c) = 11$  is prefix. The code  $C_2$  with  $C_2(a) = 0$ ,  $C_2(b) = 10$  and  $C_2(c) = 01$ , while allowing for lossless decoding, is not a prefix code since 0 is a prefix of 01. The prefix requirement is natural, and nearly ubiquitous in the data compression literature. We now explain why this is the case.

##### Example 4.1

Suppose we plan to encode a sequence of symbols  $(x_1, \dots, x_n) \in \mathcal{X}^n$ . We already designed a code  $C$  for the elements in  $\mathcal{X}$ . The natural thing to do is to encode  $(x_1, \dots, x_n)$  by the concatenated string  $C(x_1)C(x_2) \dots C(x_n)$ . In order for this method to succeed for all  $n$ , all  $(x_1, \dots, x_n) \in \mathcal{X}^n$ , the resulting procedure must define a code, i.e. the function  $C^{(n)}$  mapping  $(x_1, \dots, x_n)$  to  $C(x_1)C(x_2) \dots C(x_n)$  must be invertible. If it were not, we would have to use some marker such as a comma to separate the code words. We would then really be using a ternary rather than a binary alphabet.

Since we always want to construct codes for sequences rather than single symbols, we only allow codes  $C$  such that the extension  $C^{(n)}$  defines a code for all  $n$ . We say that such codes have ‘uniquely decodable extensions’. It is easy to see that (a) every prefix code has uniquely decodable extensions. Conversely, although this is not at all easy to see, it turns out that (b), for every code  $C$  with uniquely decodable extensions, there exists a prefix code  $C_0$  such that for all  $n$ ,  $x^n \in \mathcal{X}^n$ ,  $L_{C^{(n)}}(x^n) = L_{C_0^{(n)}}(x^n)$  [20]. Since in MDL we are only interested in code-lengths, and never in actual codes, we can restrict ourselves to prefix codes without loss of generality.

Thus, the restriction to prefix code may also be understood as a means to send concatenated messages while avoiding the need to introduce extra symbols into the alphabet.  
 $\diamond$

Whenever in the sequel we speak of ‘code’, we really mean ‘prefix code’. We call a prefix code  $C$  for a set  $\mathcal{X}$  complete if there exists no other prefix code that compresses at least one  $x$  more and no  $x$  less than  $C$ , i.e. if there exists no code  $C'$  such that for all  $x$ ,  $L_{C'}(x) \leq L_C(x)$  with strict inequality for at least one  $x$ .

#### 4.1.2 The Kraft inequality - code lengths and probabilities I

In this subsection we relate prefix codes to probability distributions. Essential for understanding the relation is the fact that no matter what code we use, most sequences cannot be compressed, as demonstrated by the following example:

##### Example 4.2 (Compression and small subsets: example 3.2 continued)

<sup>‡</sup> Also known as instantaneous codes and called, perhaps more justifiably, ‘prefix-free’ codes in [71].

In example 3.2 we featured the following three sequences:

$$00010001000100010001 \cdots 000100010001000100010001, \quad (4.1)$$

$$01110100110100100110 \cdots 1010111010111011000101100010, \quad (4.2)$$

$$00011000001010100000 \cdots 0010001000010000001000110000. \quad (4.3)$$

We showed that (a) the first sequence - an  $n$ -fold repetition of 0001 - could be substantially compressed if we use as our code a general purpose programming language (assuming that valid programs must end with a halt-statement or a closing bracket, such codes satisfy the prefix property). We also claimed that (b) the second sequence,  $n$  independent outcomes of fair coin tosses, cannot be compressed, and that (c) the third sequence could be compressed to  $\alpha n$  bits, with  $0 < \alpha < 1$ . We are now in a position to prove statement (b): strings which are 'intuitively' random cannot be substantially compressed. Let us take some arbitrary but fixed description method over the data alphabet consisting of the set of all binary sequences of length  $n$ . Such a code maps binary strings to binary strings. There are  $2^n$  possible data sequences of length  $n$ . Only two of these can be mapped to a description of length 1 (since there are only two binary strings of length 1: '0' and '1'). Similarly, only a subset of at most  $2^m$  sequences can have a description of length  $m$ . This means that at most  $\sum_{i=1}^m 2^i < 2^{m+1}$  data sequences can have a description length  $\leq m$ . The fraction of data sequences of length  $n$  that can be compressed by more than  $k$  bits is therefore at most  $2^{-k}$  and as such decreases exponentially in  $k$ . If data are generated by  $n$  tosses of a fair coin, then all  $2^n$  possibilities for the data are equally probable, so the probability that we can compress the data by more than  $k$  bits is smaller than  $2^{-k}$ . For example, the probability that we can compress the data by more than 20 bits is smaller than one in a million.

We note that after the data (4.2) has been observed, it is always possible to design a code which uses arbitrarily few bits to encode this data - the actually observed sequence may be encoded as '1' for example, and no other sequence is assigned a code word. The point is that with a code that has been designed before seeing any data, it is virtually impossible to substantially compress randomly generated data.  $\diamond$

The example demonstrates that achieving a short description length for the data is equivalent to identifying the data as belonging to a tiny, very special subset out of all *a priori* possible data sequences.

**The most important observation** Let  $\mathcal{Z}$  be finite or countable. For concreteness, we may take  $\mathcal{Z} = \{0, 1\}^n$  for some large  $n$ , say  $n = 10000$ . From example 4.2 we know that, no matter what code we use to encode values in  $\mathcal{Z}$ , 'most' outcomes in  $\mathcal{Z}$  will not be substantially compressible: at most two outcomes can have description length 1 =  $-\log \frac{1}{2}$ ; at most four outcomes can have length 2 =  $-\log \frac{1}{4}$ , and so on. Now consider any probability distribution on  $\mathcal{Z}$ . Since the probabilities  $P(z)$  must sum up to one ( $\sum_z P(z) = 1$ ), 'most' outcomes in  $\mathcal{Z}$  must have small probability in the following sense: at most 2 outcomes can have probability  $\geq \frac{1}{2}$ ; at most 4 outcomes can have probability  $\geq \frac{1}{4}$ ; at most 8 can have  $\geq \frac{1}{8}$ -th etc. This suggests an analogy between codes and probability distributions: each code induces a code length function that assigns a number to each  $z$ , where most  $z$ 's are assigned large numbers. Similarly, each distribution assigns a number to each  $z$ , where most  $z$ 's are assigned small numbers.

**Observation 4.1 (Probability mass functions correspond to code length functions)**  
 Let  $\mathcal{Z}$  be a finite or countable set and let  $P$  be a probability distribution on  $\mathcal{Z}$ . Then there exists a prefix code  $C$  for  $\mathcal{Z}$  such that for all  $z \in \mathcal{Z}$ ,  $L_C(z) = \lceil -\log P(z) \rceil$ .  $C$  is called the code corresponding to  $P$ .

Similarly, let  $C_0$  be a prefix code for  $\mathcal{Z}$ . Then there exists a (possibly defective) probability distribution  $P'$  such that for all  $z \in \mathcal{Z}$ ,  $-\log P'(z) = L_{C_0}(z)$ .  $P'$  is called the probability distribution corresponding to  $C_0$ .

Moreover,  $C_0$  is a complete prefix code if  $P$  is proper ( $\sum_z P(z) = 1$ ).

Thus, large probability according to  $P$  means small code length according to the code corresponding to  $P$  and vice versa. We are typically concerned with cases where  $\mathcal{Z}$  represents sequences of  $n$  outcomes; that is,  $\mathcal{Z} = \mathcal{X}^n$  ( $n \geq 1$ ) where  $\mathcal{X}$  is the sample space for one observation.

It turns out that this correspondence can be made mathematically precise by means of the Kraft inequality [20]. We neither precisely state nor prove this inequality; rather, in observation 4.1 we state an immediate and fundamental consequence: probability mass functions correspond to code length functions. The following example illustrates this and at the same time introduces a type of code that will be frequently employed in the sequel:

**Example 4.3 (Uniform distribution corresponds to fixed-length code)**

Suppose  $\mathcal{Z}$  has  $M$  elements. The uniform distribution  $P_U$  assigns probabilities  $\frac{1}{M}$  to each element. We can arrive at a code corresponding to  $P_U$  as follows. First, order and number the elements in  $\mathcal{Z}$  as  $0, 1, \dots, M-1$ . Then, for each  $z$  with number  $j$ , set  $C(z)$  to be equal to  $j$  represented as a binary number with  $\lceil \log M \rceil$  bits. The resulting code has, for all  $z \in \mathcal{Z}$ ,  $L_C(z) = \lceil \log M \rceil = \lceil -\log P_U(z) \rceil$ . This is a code corresponding to  $P_U$  (observation 4.1). In general, there exist several codes corresponding to  $P_U$ , one for each ordering of  $\mathcal{Z}$ . But all these codes share the same length function  $L_U(z) := \lceil -\log P_U(z) \rceil$ ; therefore,  $L_U(z)$  is the unique code length function corresponding to  $P_U$ .

For example, if  $M = 4$ ,  $\mathcal{Z} = \{a, b, c, d\}$ , we can take  $C(a) = 00$ ,  $C(b) = 01$ ,  $C(c) = 10$ ,  $C(d) = 11$  and then  $L_U(z) = 2$  for all  $z \in \mathcal{Z}$ . In general, codes corresponding to uniform distributions assign fixed lengths to each  $z$  and are called fixed-length codes. To map a non-uniform distribution to a corresponding code, we have to use a more intricate construction [20].  $\diamond$

In practical applications, we almost always deal with probability distributions  $P$  and strings  $x^n$  such that  $P(x^n)$  decreases exponentially in  $n$ ; for example, this will typically be the case if data are i.i.d., such that  $P(x^n) = \prod P(x_i)$ . Then  $-\log P(x^n)$  increases linearly in  $n$  and the effect of rounding off  $-\log P(x^n)$  becomes negligible, i.e.  $\lceil -\log P(x^n) \rceil \approx -\log P(x^n)$ . Note that the code corresponding to the product distribution of  $P$  on  $\mathcal{X}^n$  does not have to be the  $n$ -fold extension of the code for the original distribution  $P$  on  $\mathcal{X}$  - if we were to require that, the effect of rounding off

would be on the order of  $n$ . Instead, we directly design a code for the distribution on the larger space  $\mathcal{Z} = \mathcal{X}^n$ . In this way, the effect of rounding changes the code length by at most 1 bit, which is truly negligible. For this and other<sup>§</sup> reasons, we henceforth simply neglect the integer requirement for code lengths. This simplification allows us to identify code length functions and (defective) probability mass functions, such that a short code length corresponds to a high probability and vice versa. Furthermore, as we will see, in MDL we are not interested in the details of actual encodings  $C(z)$ ; we only care about the code lengths  $L_{C(z)}$ . It is so useful to think about these as log-probabilities, and conveniently dispense with integer restrictions and probabilities, that we will simply redefine prefix code length functions as (defective) probability mass functions that can have non-integer code lengths - see observation 4.2.

**Observation 4.2 (New definition of code length function)**

*In MDL we are NEVER concerned with actual encodings; we are only concerned with code length functions. The set of all code length functions for finite or countable sample space  $\mathcal{Z}$  is defined as:*

$$\mathcal{L}_{\mathcal{Z}} = \{L : \mathcal{Z} \rightarrow [0, \infty] \mid \sum_{z \in \mathcal{X}} 2^{-L(z)} \leq 1\}, \quad (4.4)$$

or equivalently,  $\mathcal{L}_{\mathcal{Z}}$  is the set of those functions  $L$  on  $\mathcal{Z}$  such that there exists a function  $Q$  with  $\sum_z Q(z) \leq 1$  and for all  $z$ ,  $L(z) = -\log Q(z)$ . ( $Q(z) = 0$  corresponds to  $L(z) = \infty$ ). Again,  $\mathcal{Z}$  usually represents a sample of  $n$  outcomes:  $\mathcal{Z} = \mathcal{X}^n$  ( $n \leq 1$ ) where  $\mathcal{X}$  is the sample space for one observation.

The following example illustrates idealized code length functions and at the same time introduces a type of code that will be frequently used in the sequel:

**Example 4.4 ('Almost' uniform code for the positive integers)**

Suppose we want to encode a number  $k \in \{1, 2, \dots\}$ . In example 4.3, we saw that in order to encode a number between 1 and  $M$ , we need  $\log M$  bits. What if we cannot determine the maximum  $M$  in advance? We cannot just encode  $k$  using the uniform code for  $\{1, \dots, k\}$ , since the resulting code would not be prefix. So in general, we will need more than  $\log k$  bits. Yet there exists a prefix-free code which performs 'almost' as well as  $\log k$ . The simplest of such codes works as follows.  $k$  is described by a code word starting with  $\lceil \log k \rceil$  0s. This is followed by a 1, and then  $k$  is encoded using the uniform code for  $\{1, \dots, 2^{\lceil \log k \rceil}\}$ . With this protocol, a decoder can first reconstruct  $\lceil \log k \rceil$  by counting all 0's before the leftmost 1 in the encoding. He then has an upper bound on  $k$  and can use this knowledge to decode  $k$  itself. This protocol uses less than  $2\lceil \log k \rceil + 1$  bits. Working with idealized, non-integer code-lengths we can simplify this to  $2\log k + 1$  bits. To see this, consider the function  $P(x) = 2^{-2\log x - 1}$ . An easy calculation gives

$$\sum_{x \in \{1, 2, \dots\}} P(x) = \sum_{x \in \{1, 2, \dots\}} 2^{-2\log x - 1} = \frac{1}{2} \sum_{x \in \{1, 2, \dots\}} x^{-2} < \frac{1}{2} + \frac{1}{2} \sum_{x \in \{2, 3, \dots\}} \frac{1}{x(x-1)} \leq 1,$$

<sup>§</sup> For example, with non-integer code lengths the notion of 'code' becomes invariant to the size of the alphabet in which we describe data.

so that  $P$  is a (defective) probability distribution. Thus, by our new definition (observation 4.2), there exists a prefix code with, for all  $k$ ,  $L(k) = -\log P(k) = 2 \log k + 1$ . We call the resulting code the ‘simple standard code for the integers’. In section 4.4 we will see that it is an instance of a so-called ‘universal’ code.

The idea can be refined to lead to codes with lengths  $\log k + \mathcal{O}(\log \log k)$ ; the ‘best’ possible refinement, with code lengths  $L(k)$  increasing monotonically but as slowly as possible in  $k$ , is known as ‘the universal code for the integers’ [89]. However, for our purposes in this chapter, it is good enough to encode integers  $k$  with  $2 \log k + 1$  bits.  $\diamond$

#### Example 4.5 (Example 3.2 and 4.2, continued)

We are now also in a position to prove the third and final claim of examples 3.2 and 4.2. Consider the three sequences (4.1), (4.2) and (4.3) again. It remains to investigate how much the third sequence can be compressed. Assume for concreteness that, before seeing the sequence, we are told that the sequence contains a fraction of 1s equal to  $\frac{1}{5} + \epsilon$  for some small unknown  $\epsilon$ . By the Kraft inequality, observation 4.1, for all distributions  $P$ , there exists some code on sequences of length  $n$  such that for all  $x^n \in \mathcal{X}^n$ ,  $L(x^n) = \lceil -\log P(x^n) \rceil$ . The fact that the fraction of 1s is approximately equal to  $\frac{1}{5}$  suggests to model  $x^n$  as independent outcomes of a coin with bias  $\frac{1}{5}$ -th. The corresponding distribution  $P_0$  satisfies

$$\begin{aligned} -\log P_0(x^n) &= -\log \left(\frac{1}{5}\right)^{n_{[1]}} \left(\frac{4}{5}\right)^{n_{[0]}} = n\left(-\left(\frac{1}{5} + \epsilon\right) \log \frac{1}{5} - \left(\frac{4}{5} - \epsilon\right) \log \frac{4}{5}\right) \\ &= n\left(\log 5 - \frac{8}{5} + 2\epsilon\right), \end{aligned}$$

where  $n_{[j]}$  denotes the number of occurrences of symbol  $j$  in  $x^n$ . For small enough  $\epsilon$ , the part between brackets is smaller than 1, so that, using the code  $L_0$  with lengths  $-\log P_0$ , the sequence can be encoded using  $\alpha n$  bits where  $\alpha$  satisfies  $0 < \alpha < 1$ . Thus, using the code  $L_0$ , the sequence can be compressed by a linear amount, if we use a specially designed code that assigns short code lengths to sequences with about four times as many 0s than 1s.

We note that after the data (4.3) has been observed, it is always possible to design a code which uses arbitrarily few bits to encode  $x^n$  - the actually observed sequence may be encoded as ‘1’ for example, and no other sequence is assigned a code word. The point is that with a code that has been designed before seeing the actual sequence, given only the knowledge that the sequence will contain approximately four times as many 0s than 1s, the sequence is guaranteed to be compressed by an amount linear in  $n$ .  $\diamond$

**Continuous sample spaces** How does the correspondence work for continuous valued  $\mathcal{X}$ ? In this chapter we only consider  $P$  on  $\mathcal{X}$  such that  $P$  admits a density<sup>†</sup>. Whenever in the following we make a general statement about sample spaces  $\mathcal{X}$  and distributions  $P$ ,  $\mathcal{X}$  may be finite, countable or any subset of  $\mathbb{R}^l$ , for any integer  $l \geq 1$ , and  $P(x)$  represents the probability mass function or density of  $P$ , as the case may be. In the continuous case, all sums should be read as integrals. The correspondence between probability distributions and codes may be extended to distributions on continuous-valued  $\mathcal{X}$ : we may think of  $L(x^n) := -\log P(x^n)$  as a code-length function corresponding

<sup>†</sup> As understood in elementary probability, i.e. with respect to the Lebesgue measure.

to  $Z = \mathcal{X}^n$  encoding the values in  $\mathcal{X}^n$  at unit precision; here  $P(x^n)$  is the density of  $x^n$  according to  $P$ . We refer to [20] for further details.

#### 4.1.3 The information inequality - code lengths and probabilities II

In the previous subsection, we established the first fundamental relation between probability distributions and code length functions. We now discuss the second relation, which is nearly as important.

In the correspondence to code length functions, probability distributions were treated as mathematical objects and nothing else. That is, if we decide to use a code  $C$  to encode our data, this definitely does not necessarily mean that we assume our data to be drawn according to the probability distribution corresponding to  $L$ : we may have no idea what distribution generates our data; or conceivably, such a distribution may not even exist<sup>||</sup>. Nevertheless, if the data are distributed according to some distribution  $P$ , then the code corresponding to  $P$  turns out to be the optimal code to use, in an expected sense - see observation 4.3. This result may be recast as follows: for all distributions  $P$  and  $Q$  with  $Q \neq P$ ,

$$E_P(-\log Q(X)) > E_P(-\log P(X)).$$

In this form, the result is known as the information inequality. It is easily proved using concavity of the logarithm [20].

**Observation 4.3 (The  $P$  that corresponds to  $L$  minimizes expected code length)**

Let  $P$  be a distribution on (finite, countable or continuous-valued)  $Z$  and let  $L$  be defined by

$$L := \min_{L \in \mathcal{L}_Z} E_P(L(Z)). \quad (4.5)$$

Then  $L$  exists, is unique, and is identical to the code length function corresponding to  $P$ , with lengths  $L(z) = -\log P(z)$ .

The information inequality says the following: suppose  $Z$  is distributed according to  $P$  ('generated by  $P$ '). Then, among all possible codes for  $Z$ , the code with lengths  $-\log P(Z)$  'on average' gives the shortest encodings of outcomes of  $P$ . Why should we be interested in the average? The law of large numbers [30] implies that, for large samples of data distributed according to  $P$ , with high  $P$ -probability, the code that gives the shortest expected lengths will also give the shortest actual code lengths, which is what we are really interested in. This will hold if data are i.i.d., but also more generally if  $P$  defines a 'stationary and ergodic' process.

#### Example 4.6

Let us briefly illustrate this. Let  $P^*$ ,  $Q_A$  and  $Q_B$  be three probability distributions on  $\mathcal{X}$ , extended to  $Z = \mathcal{X}^n$  by independence. Hence  $P^*(x^n) = \prod P^*(x_i)$  and similarly

<sup>||</sup> Even if one adopts a Bayesian stance and postulates that an agent can come up with a (subjective) distribution for every conceivable domain, this problem remains: in practice, the adopted distribution may be so complicated that we cannot design the optimal code corresponding to it, and have to use some ad hoc one instead.

for  $Q_A$  and  $Q_B$ . Suppose we obtain a sample generated by  $P^*$ . A and B both want to encode the sample using as few bits as possible, but neither knows that  $P^*$  has actually been used to generate the sample. A decides to use the code corresponding to distribution  $Q_A$  and B decides to use the code corresponding to  $Q_B$ . Suppose that  $E_{P^*}(-\log Q_A(X)) < E_{P^*}(-\log Q_B(X))$ . Then, by the law of large numbers, with  $P^*$ -probability 1,  $n^{-1}(-\log Q_j(X_1, \dots, X_n)) \rightarrow E_{P^*}(-\log Q_j(X))$ , for both  $j \in \{A, B\}$  (note  $-\log Q_j(X^n) = -\sum_{i=1}^n \log Q_j(X_i)$ ). It follows that, with probability 1, A will need less (linearly in  $n$ ) bits to encode  $X_1, \dots, X_n$  than B.  $\diamond$

The qualitative content of this result is not so surprising: in a large sample generated by  $P$ , the frequency of each  $x \in \mathcal{X}$  will be approximately equal to the probability  $P(x)$ . In order to obtain a short code length for  $x^n$ , we should use a code that assigns a small code length to those symbols in  $\mathcal{X}$  with high frequency (probability), and a large code length to those symbols in  $\mathcal{X}$  with low frequency (probability).

**Summary** In this section we introduced (prefix) codes and thoroughly discussed the relation between probabilities and code lengths. We are now almost ready to formalize a simple version of MDL - but first we need to review some concepts of statistics.

## 4.2 Statistical preliminaries and example models

In the next section, we formally introduce the crude form of MDL. We will freely use some convenient statistical concepts which we review in this section; for details see, for example, [15]. We also describe the model class of Markov chains of arbitrary order, which we use as our running example. These admit a simpler treatment than the polynomials, to which we return in section 4.7.

**Statistical preliminaries** A probabilistic model\*  $\mathcal{M}$  is a set of probabilistic sources. Typically one uses the word ‘model’ to denote sources of the same functional form. We often index the elements  $P$  of a model  $\mathcal{M}$  using some parameter  $\theta$ . In that case we write  $P$  as  $P(\cdot|\theta)$ , and  $\mathcal{M}$  as  $\mathcal{M} = \{P(\cdot|\theta) | \theta \in \Theta\}$ , for some parameter space  $\Theta$ . If  $\mathcal{M}$  can be parameterized by some connected  $\Theta \subseteq \mathbb{R}^k$  for some  $k \geq 1$  and the mapping  $\theta \rightarrow P(\cdot|\theta)$  is smooth (appropriately defined), we call  $\mathcal{M}$  a parametric model or family. For example, the model  $\mathcal{M}$  of all normal distributions on  $\mathcal{X} = \mathbb{R}$  is a parametric model that can be parameterized by  $\theta = (\mu, \sigma^2)$  where  $\mu$  is the mean and  $\sigma^2$  is the variance of the distribution indexed by  $\theta$ . The family of all Markov chains of all orders is a model, but not a parametric model. We call a model  $\mathcal{M}$  an i.i.d. model if, according to all  $P \in \mathcal{M}$ ,  $X_1, X_2, \dots$  are i.i.d. We call  $\mathcal{M}$   $k$ -dimensional if  $k$  is the smallest integer  $k$  so that  $\mathcal{M}$  can be smoothly parameterized by some  $\Theta \subseteq \mathbb{R}^k$ .

For a given model  $\mathcal{M}$  and sample  $D = x^n$ , the maximum likelihood (ML)  $P$  is the  $P \in \mathcal{M}$  maximizing  $P(x^n)$ . For a parametric model with parameter space  $\Theta$ , the maximum likelihood estimator  $\hat{\theta}$  is the function that, for each  $n$ , maps  $x^n$  to the  $\theta \in \Theta$  that maximizes the likelihood  $P(x^n|\theta)$ . The ML estimator may be viewed as a ‘learning algorithm’. This is a procedure that, when getting input a sample  $x^n$  of arbitrary length, outputs a parameter or hypothesis  $P_n \in \mathcal{M}$ . We say that a learning algorithm

\* Henceforth, we simply use ‘model’ to denote probabilistic models; we typically use  $\mathcal{H}$  to denote sets of hypotheses such as polynomials, and reserve  $\mathcal{M}$  for probabilistic models.

is consistent relative to distance measure  $d$ , if for all  $P^* \in \mathcal{M}$ , with data distributed according to  $P^*$ , then the output  $P_n$  converges to  $P^*$  in the sense that  $d(P^*, P_n) \rightarrow 0$  with  $P^*$ -probability 1. Thus, if  $P^*$  is the ‘true’ state of nature, then given enough data, the learning algorithm will learn a good approximation of  $P^*$  with very high probability.

**Example 4.7 (Markov and Bernoulli models)**

Recall that a  $k$ -th order Markov chain on  $\mathcal{X} = \{0, 1\}$  is a probabilistic source such that for every  $n > k$ ,

$$\begin{aligned} P(X_n = 1 | X_{n-1} = x_{n-1}, \dots, X_{n-k} = x_{n-k}) = \\ P(X_n = 1 | X_{n-1} = x_{n-1}, \dots, X_{n-k} = x_{n-k}, \dots, X_1 = x_1). \end{aligned} \quad (4.6)$$

That is, the probability distribution on  $X_n$  depends only on the  $k$  symbols preceding  $n$ . Thus, there are  $2^k$  possible distributions of  $X_n$ , and each such distribution is identified with a state of the Markov chain. To fully identify the chain, we also need to specify the starting state, defining the first  $k$  outcomes  $X_1, \dots, X_k$ . The  $k$ -th order Markov model is the set of all  $k$ -th order Markov chains, i.e., all sources satisfying (4.6) equipped with a starting state.

The special case of the 0-th order Markov model is the Bernoulli or biased coin model, which we denote by  $\mathcal{B}^{(0)}$ . We can parameterize the Bernoulli model by a parameter  $\theta \in [0, 1]$  representing the probability of observing a 1. Thus,  $\mathcal{B}^{(0)} = \{P(\cdot|\theta) | \theta \in [0, 1]\}$ , with  $P(x^n|\theta)$  by definition equal to

$$P(x^n|\theta) = \prod_{i=1}^n P(x_i|\theta) = \theta^{n_{[1]}} (1 - \theta)^{n_{[0]}},$$

where  $n_{[1]}$  stands for the number of 1s, and  $n_{[0]}$  for the number of 0s in the sample. Note that the Bernoulli model is i.i.d. and that  $n_{[1]} + n_{[0]} = n$ . The log-likelihood is given by

$$\log P(x^n|\theta) = n_{[1]} \log \theta + n_{[0]} \log(1 - \theta). \quad (4.7)$$

Taking the derivative of (4.7) with respect to  $\theta$ , we see that for fixed  $x^n$ , the log-likelihood is maximized by setting the probability of 1 equal to the observed frequency. Since the logarithm is a monotonically increasing function, the likelihood is maximized at the same value: the ML estimator is given by  $\hat{\theta}(x^n) = n_{[1]}/n$ .

Similarly, the first-order Markov model  $\mathcal{B}^{(1)}$  can be parameterized by a vector  $\theta = (\theta_{[1|0]}, \theta_{[1|1]}) \in [0, 1]^2$  together with a starting state in  $\{0, 1\}$ . Here  $\theta_{[1|j]}$  represents the probability of observing a 1 following the symbol  $j$ . The log-likelihood is given by

$$\begin{aligned} \log P(x^n|\theta) = n_{[1|1]} \log \theta_{[1|1]} + n_{[0|1]} \log(1 - \theta_{[1|1]}) \\ + n_{[1|0]} \log \theta_{[1|0]} + n_{[0|0]} \log(1 - \theta_{[1|0]}), \end{aligned}$$

$n_{[i|j]}$  denoting the number of times outcome  $i$  is observed in state (previous outcome)  $j$ . This is maximized by setting  $\hat{\theta} = (\hat{\theta}_{[1|0]}, \hat{\theta}_{[1|1]})$ , with  $\hat{\theta}_{[i|j]} = n_{[i|j]}/n_{[j]}$  set to the conditional frequency of  $i$  preceded by  $j$ . In general, a  $k$ -th order Markov chain has  $2^k$  parameters and the corresponding likelihood is maximized by setting the parameter  $\theta_{[i|j]}$  equal to the number of times  $i$  was observed in state  $j$  divided by the number of times the chain was in state  $j$ .  $\diamond$



Suppose now we are given data  $D = x^n$  and we want to find the Markov chain that best explains  $D$ . Since we do not want to restrict ourselves to chains of fixed order, we run a large risk of overfitting: simply picking, among all Markov chains of each order, the ML Markov chain that maximizes the probability of the data, we typically end up with a chain of order  $n - 1$  with starting state given by the sequence  $x_1, \dots, x_{n-1}$ , and  $P(X_n = x_n | X_{n-1} = x_{n-1}) = 1$ . Such a chain will assign probability 1 to  $x^n$ . Below we show that MDL makes a more reasonable choice.

### 4.3 Crude MDL

Based on the information theoretic (section 4.1) and statistical (section 4.2) preliminaries discussed before, we now formalize a first, crude version of MDL.

Let  $\mathcal{M}$  be a class of probabilistic sources (not necessarily Markov chains). Suppose we observe a sample  $D = (x_1, \dots, x_n) \in \mathcal{X}^n$ . Recall ‘the crude\* two-part code MDL principle’ from section 3.3:

#### Crude, two-part version of MDL principle

Let  $\mathcal{H}^{(1)}, \mathcal{H}^{(2)}, \dots$  be a set of candidate models. The best point hypothesis  $H \in \mathcal{H}^{(1)} \cup \mathcal{H}^{(2)} \cup \dots$  to explain data  $D$  is the one which minimizes the sum  $L(H) + L(D|H)$ , where

- $L(H)$  is the length, in bits, of the description of the hypothesis, and
- $L(D|H)$  is the length, in bits, of the description of the data when encoded with the help of the hypothesis.

The best model to explain  $D$  is the smallest model containing the selected  $H$ .

In this section, we implement this crude MDL principle by giving a precise definition of the terms  $L(H)$  and  $L(D|H)$ . To make the first term precise, we must design a code  $C_1$  for encoding hypotheses  $H$  such that  $L(H) = L_{C_1}(H)$ . For the second term, we must design a set of codes  $C_{2,H}$  (one for each  $H \in \mathcal{M}$ ) such that for all  $D \in \mathcal{X}^n$ ,  $L(D|H) = L_{C_{2,H}}(D)$ . We start by describing the codes  $C_{2,H}$ .

#### 4.3.1 Description length of data given hypotheses

Given a sample of size  $n$ , each hypothesis  $H$  may be viewed as a probability distribution on  $\mathcal{X}^n$ . We denote the corresponding probability mass function by  $P(\cdot|H)$ . We need to associate with  $P(\cdot|H)$  a code, or really, just a code length function for  $\mathcal{X}^n$ . We already know that there exists a code with length function  $L$  such that for all  $x^n \in \mathcal{X}^n$ ,  $L(x^n) = -\log P(x^n|H)$ . This is the code that we will pick. It is a natural choice for two reasons:

1. With this choice, the code length  $L(x^n|H)$  is equal to minus the log-likelihood of  $x^n$  according to  $H$ , which is a standard statistical notion of ‘goodness-of-fit’.

\* The terminology ‘crude MDL’ is not standard. It is introduced here for pedagogical reasons, to clarify the importance of having a single, unified principle for designing codes. It should be noted that Rissanen’s and Barron’s early theoretical papers on MDL already contain such principles, albeit in a slightly different form than in their recent papers. Early practical applications [83], [43] often do use ad hoc two-part codes which really are ‘crude’ in the sense defined here.

2. If the data turn out to be distributed according to  $P$ , then the code  $L(\cdot|H)$  will uniquely minimize the expected code length (section 4.1).

The second item implies that our choice is, in a sense, the only reasonable choice<sup>†</sup>. To see this, suppose  $\mathcal{M}$  is a finite i.i.d. model containing, say,  $M$  distributions. Suppose we assign an arbitrary but finite code length  $L(H)$  to each  $H \in \mathcal{M}$ . Suppose  $X_1, X_2, \dots$  are actually distributed i.i.d. according to some ‘true’  $H^* \in \mathcal{M}$ . By the reasoning of example 4.6, we see that MDL will select the true distribution  $P(\cdot|H^*)$  for all large  $n$ , with probability 1. This means that MDL is consistent for finite  $\mathcal{M}$ . If we were to assign codes to distributions in some other manner not satisfying  $L(D|H) = -\log P(D|H)$ , then there would exist distributions  $P(\cdot|H)$  such that  $L(D|H) \neq -\log P(D|H)$ . But by observation 4.1, there must be some distribution  $P(\cdot|H')$  with  $L(\cdot|H') = -\log P(\cdot|H')$ . Now let  $\mathcal{M} = \{H, H'\}$  and suppose data are distributed according to  $P(\cdot|H')$ . Then, by the reasoning of example 4.6, MDL would select  $H$  rather than  $H'$  for all large  $n$ ! Thus, MDL would be inconsistent even in this simplest of all imaginable cases - there would then be no hope for good performance in the considerably more complex situations we intend to use it for<sup>‡</sup>.

### 4.3.2 Description length of hypotheses

In its weakest and crudest form, the two-part code MDL principle does not give any guidelines as to how to encode hypotheses (probability distributions). Every code for encoding hypotheses is allowed, as long as such a code does not change with the sample size  $n$ .

To see the danger in allowing codes to depend on  $n$ , consider the Markov chain example: if we were allowed to use different codes for different  $n$ , we could use, for each  $n$ , a code assigning a uniform distribution to all Markov chains of order  $n - 1$  with all parameters equal to 0 or 1. Since there are only a finite number ( $2^{n-1}$ ) of these, this is possible. But then, for each  $n$ ,  $x^n \in \mathcal{X}^n$ , MDL would select the ML Markov chain of order  $n - 1$ . Thus, MDL would coincide with ML and, no matter how large  $n$ , we would overfit.

**Consistency of two-part MDL** Remarkably, if we fix an arbitrary code for all hypotheses, identical for all sample sizes  $n$ , this is sufficient to make MDL consistent<sup>§</sup> for a wide variety of models, including the Markov chains. For example, let  $L$  be the length function corresponding to some code for the Markov chains. Suppose some Markov chain  $P^*$  generates the data such that  $L(P^*) < 1$  under our coding scheme. Then, loosely speaking, for every  $P^*$  of every order, with probability 1 there exists some  $n_0$  such that for all samples larger than  $n_0$ , two-part MDL will select  $P^*$  - here  $n_0$  may depend on  $P^*$  and  $L$ .

While this result indicates that MDL may be doing something sensible, it certainly does not justify the use of arbitrary codes - different codes will lead to preferences of

<sup>†</sup> But see chapter 6 for more discussion.

<sup>‡</sup> See section 3.5 of chapter 3 for a discussion on the role of consistency in MDL.

<sup>§</sup> See, for example [8], [6].

different hypotheses, and it is not at all clear how a code should be designed that leads to good inferences with small, practically relevant sample sizes.

Barron and Cover [8] have developed a precise theory of how to design codes  $C_1$  in a 'clever' way, anticipating the developments of 'refined MDL'. Practitioners have often simply used 'reasonable' coding schemes, based on the following idea. Usually there exists some 'natural' decomposition of the models under consideration,  $\mathcal{M} = \bigcup_{k>0} \mathcal{M}^{(k)}$  where the dimension of  $\mathcal{M}^{(k)}$  grows with  $k$  but is not necessarily equal to  $k$ . In the Markov chain example, we have  $\mathcal{B} = \bigcup \mathcal{B}^{(k)}$  where  $\mathcal{B}^{(k)}$  is the  $k$ -th order,  $2^k$ -parameter Markov model. Then within each submodel  $\mathcal{M}^{(k)}$ , we may use a fixed-length code for  $\theta \in \Theta^{(k)}$ . Since the set  $\Theta^{(k)}$  is typically a continuum, we somehow need to discretize it to achieve this.

#### Example 4.8 (A very crude code for the Markov chains)

We can describe a Markov chain of order  $k$  by first describing  $k$ , and then describing a parameter vector  $\theta \in [0, 1]^{k'}$  with  $k' = 2^k$ . We describe  $k$  using our simple code for the integers (example 4.4). This takes  $2 \log k + 1$  bits. We now have to describe the  $k'$ -component parameter vector. We saw in example 4.7 that for any  $x^n$ , the best-fitting (ML)  $k$ -th order Markov chain can be identified with  $k'$  frequencies. It is not hard to see that these frequencies are uniquely determined by the counts  $n_{[1|0\dots 00]}, n_{[1|0\dots 01]}, \dots, n_{[1|1\dots 11]}$ . Each individual count must be in the  $(n+1)$ -element set  $\{0, 1, \dots, n\}$ . Since we assume  $n$  is given in advance<sup>†</sup>, we may use a simple fixed-length code to encode this count, taking  $\log(n+1)$  bits (example 4.3). Thus, once  $k$  is fixed, we can describe such a Markov chain by a uniform code using  $k' \log(n+1)$  bits. With the code just defined we get for any  $P \in \mathcal{B}$ , indexed by parameter  $\Theta^{(k)}$ ,

$$L(P) = L(k, \Theta^{(k)}) = 2 \log k + 1 + k \log(n+1),$$

so that with these codes, MDL tells us to pick the  $k, \theta^{(k)}$  minimizing

$$L(k, \theta^{(k)}) + L(D|k, \theta^{(k)}) = 2 \log k + 1 + k \log(n+1) - \log P(D|k, \theta^{(k)}), \quad (4.8)$$

where the  $\theta^{(k)}$  that is chosen will be equal to the ML estimator for  $\mathcal{M}^{(k)}$ .  $\diamond$

**Why not this code?** We may ask two questions about this code. First, why did we only reserve code words for  $\theta$  that are potentially ML estimators for the given data? The reason is that, given  $k' = 2^k$ , the code length  $L(D|k, \theta^{(k)})$  is minimized by  $\hat{\theta}^{(k)}(D)$ , the ML estimator within  $\theta^{(k)}$ . Reserving code words for  $\theta \in [0, 1]^{k'}$  that cannot be ML estimates would only serve to lengthen  $L(D|k, \theta^{(k)})$  and can never shorten  $L(k|\theta^{(k)})$ . Thus, the total description length needed to encode  $D$  will increase. Since our stated goal is to minimize description lengths, this is undesirable.

<sup>†</sup> Strictly speaking, the assumption that  $n$  is given in advance (i.e., both encoder and decoder know  $n$ ) contradicts the earlier requirement that the code to be used for encoding hypotheses is not allowed to depend on  $n$ . Thus, we should first encode some  $n$  explicitly, using  $2 \log n + 1$  bits (example 4.4), and then pick the  $n$  (typically, but not necessarily equal to the actual sample size) that allows for the shortest three-part code length of the data (first encode  $n$ , then  $(k, \theta)$ , then the data). In practice this will not significantly alter the chosen hypothesis, unless for some quite special data sequences.

However, by the same logic we may also ask whether we have not reserved too many code words for  $\theta \in [0, 1]^{k'}$ . And in fact, it turns out that we have: the distance between two adjacent ML estimators is  $\mathcal{O}(\frac{1}{n})$ . Indeed, if we had used a coarser precision, only reserving code words for parameters with distances  $\mathcal{O}(\frac{1}{\sqrt{n}})$ , we would obtain smaller code lengths - (4.8) would become

$$L(k, \theta^{(k)}) + L(D|k, \theta^{(k)}) = -\log P(D|k, \hat{\theta}^{(k)}) + \frac{k}{2} \log n + c_k, \quad (4.9)$$

where  $c_k$  is a small constant depending on  $k$ , but not on  $n$  [8]. In section 4.5 we show that (4.9) is in some sense ‘optimal’.

**The good news and the bad news** The good news is (1) we have found a principled, non-arbitrary manner to encode data  $D$  given a probability distribution  $H$ , namely, to use the code with lengths  $-\log P(D|H)$ ; and (2), asymptotically, any code for hypotheses will lead to a consistent criterion. The bad news is that we have not found clear guidelines to design codes for hypotheses  $H \in \mathcal{M}$ . We found some intuitively reasonable codes for Markov chains, and we then reasoned that these could be somewhat ‘improved’, but what is conspicuously lacking is a sound theoretical principle for designing and improving codes.

We take the good news to mean that our idea may be worth pursuing further. We take the bad news to mean that we do have to modify or extend the idea to get a meaningful, non-arbitrary and practically relevant model selection method. Such an extension was already suggested in Rissanen’s early works [88], [89] and refined by Barron and Cover [8]. However, in these works, the principle was still restricted to two-part codes. To get a fully satisfactory solution, we need to move to ‘universal codes’, of which the two-part codes are merely a special case.

## 4.4 Information theory II: universal codes and models

We have just indicated why the two-part code formulation of MDL needs to be refined. It turns out that the key concept we need is that of universal coding. Loosely speaking, a code  $\bar{L}$  that is universal relative to a set of candidate codes  $\mathcal{L}$  allows us to compress every sequence  $x^n$  almost as well as the code in  $\mathcal{L}$  that compresses that particular sequence most. Two-part codes are universal (section 4.4.1), but there exist other universal codes such as the Bayesian mixture code (section 4.4.2) and the normalized maximum likelihood (NML) code (section 4.4.3). We also discuss universal models, which are just the probability distributions corresponding to universal codes. In this section, we are not concerned with learning from data; we only care about compressing data as much as possible. We connect our findings with learning in section 4.5.

**Coding as communication** Like many other topics in coding, ‘universal coding’ can best be explained if we think of descriptions as messages: we can always view a description as a message that some sender or encoder, say A, sends to some receiver or decoder, say B. Before sending any messages, A and B meet in person. They agree on the set of messages that A may send to B. Typically, this will be the set  $\mathcal{X}^n$  of sequences  $x_1, \dots, x_n$ , where each  $x_i$  is an outcome in the space  $\mathcal{X}$ . They also agree upon a (prefix) code that will be used by A to send his messages to B. Once this has

been done, A and B go back to their respective homes and A sends his messages to B in the form of binary strings. The unique decodability property of prefix codes implies that, when B receives a message, he should always be able to decode it in a unique manner.

**Universal coding** Suppose our encoder/sender is about to observe a sequence  $x^n \in \mathcal{X}^n$  which he plans to compress as much as possible. Equivalently, he wants to send an encoded version of  $x^n$  to the receiver using as few bits as possible. Sender and receiver have a set of candidate codes  $\mathcal{L}$  for  $\mathcal{X}^n$  available\*. They believe or hope that one of these codes will allow for substantial compression of  $x^n$ . However, they must decide on a code for  $\mathcal{X}^n$  before sender observes the actual  $x^n$ , and they do not know which code in  $\mathcal{L}$  will lead to good compression of the actual  $x^n$ . What is the best thing they can do? They may be tempted to try the following: upon seeing  $x^n$ , sender simply encodes/sends  $x^n$  using the  $L \in \mathcal{L}$  that minimizes  $L(x^n)$  among all  $L \in \mathcal{L}$ . But this naive scheme will not work: since decoder/receiver does not know what  $x^n$  has been sent before decoding the message, he does not know which of the codes in  $\mathcal{L}$  has been used by sender/encoder. Therefore, decoder cannot decode the message: the resulting protocol does not constitute a uniquely decodable, let alone a prefix code. Indeed, as we show below, in general no code  $\bar{L}$  exists such that for all  $x^n \in \mathcal{X}^n$ ,  $\bar{L}(x^n) \leq \min_{L \in \mathcal{L}} L(x^n)$ : in words, there exists no code which, no matter what  $x^n$  is, always mimics the best code for  $x^n$ .

#### Example 4.9

Suppose we think that our sequence can be reasonably well-compressed by a code corresponding to some biased coin model. For simplicity, we restrict ourselves to a finite number of such models. Thus, let  $\mathcal{L} = \{L_1, \dots, L_9\}$  where  $L_1$  is the code length function corresponding to the Bernoulli model  $P(\cdot|\theta)$  with parameter  $\theta = 0.1$ ,  $L_2$  corresponds to  $\theta = 0.2$  and so on. From (4.7) we see that, for example,

$$\begin{aligned} L_8(x^n) &= -\log P(x^n|0.8) = -n_{[0]} \log 0.2 - n_{[1]} \log 0.8 \\ L_9(x^n) &= -\log P(x^n|0.9) = -n_{[0]} \log 0.1 - n_{[1]} \log 0.9. \end{aligned}$$

Both  $L_8(x^n)$  and  $L_9(x^n)$  are linearly increasing in the number of 1s in  $x^n$ . However, if the frequency  $\frac{n_1}{n}$  is approximately 0.8, then  $\min_{L \in \mathcal{L}} L(x^n)$  will be achieved for  $L_8$ . If  $\frac{n_1}{n} \approx 0.9$  then  $\min_{L \in \mathcal{L}} L(x^n)$  is achieved for  $L_9$ . More generally, if  $\frac{n_1}{n} \approx \frac{j}{10}$  then  $L_j$  achieves the minimum†. We would like to send  $x^n$  using a code  $\bar{L}$  such that for all  $x^n$ , we need at most  $\hat{L}(x^n)$  bits, where  $\hat{L}(x^n)$  is defined as  $\hat{L}(x^n) := \min_{L \in \mathcal{L}} L(x^n)$ . Since  $-\log$  is monotonically decreasing,  $\hat{L}(x^n) = -\log P(x^n|\hat{\theta}(x^n))$ . We already gave an informal explanation why a code with lengths  $\hat{L}$  does not exist. We can now explain this more formally as follows: if such a code were to exist, it would correspond to some distribution  $\bar{P}$ . Then we would have for all  $x^n$ ,  $\bar{L}(x^n) = -\log \bar{P}(x^n)$ .

\* As explained in observation 4.2, we identify these codes with their length functions, which is the only aspect we are interested in.

† The reason is that, in the full Bernoulli model with parameter  $\theta \in [0, 1]$ , the maximum likelihood estimator is given by  $\frac{n_1}{n}$ , see example 4.7. Since the likelihood  $\log P(x^n|\theta)$  is a continuous function of  $\theta$ , this implies that if the frequency  $\frac{n_1}{n}$  in  $x^n$  is approximately (but not precisely)  $\frac{j}{10}$ , then the ML estimator in the restricted model  $\{0.1, \dots, 0.9\}$  is still given by  $\hat{\theta} = \frac{j}{10}$ . Then  $\log P(x^n|\theta)$  is maximized by  $\hat{\theta} = \frac{j}{10}$ , so that the  $L \in \mathcal{L}$  that minimizes code length corresponds to  $\theta = \frac{j}{10}$ .

But, by definition, for all  $x^n \in \mathcal{X}^n$ ,  $\bar{L}(x^n) \leq \hat{L}(x^n) = -\log P(x^n|\hat{\theta}(x^n))$  where  $\hat{\theta}(x^n) \in \{0.1, \dots, 0.9\}$ . Thus we get for all  $x^n$ ,  $-\log \bar{P}(x^n) \leq -\log P(x^n|\hat{\theta}(x^n))$  or  $\bar{P}(x^n) \geq P(x^n|\hat{\theta}(x^n))$ , so that, since  $|\mathcal{L}| > 1$ ,

$$\sum_{x^n} \bar{P}(x^n) \geq \sum_{x^n} P(x^n|\hat{\theta}(x^n)) = \sum_{x^n} \max_{\theta} P(x^n|\theta) > 1, \quad (4.10)$$

where the last inequality follows because for any two  $\theta_1, \theta_2$  with  $\theta_1 \neq \theta_2$ , there is at least one  $x^n$  with  $P(x^n|\theta_1) > P(x^n|\theta_2)$ . Equation (4.10) says that  $\bar{P}$  is not a probability distribution. It follows that  $\bar{L}$  cannot be a code length function. The argument can be extended beyond the Bernoulli model of the example above: as long as  $|\mathcal{L}| > 1$ , and all codes in  $\mathcal{L}$  correspond to a non-defective distribution, (4.10) must still hold, so that there exists no code  $\bar{L}$  with  $\bar{L}(x^n) = \hat{L}(x^n)$  for all  $x^n$ . The underlying reason that no such code exists is the fact that probabilities must sum up to something  $\leq 1$ ; or equivalently, that there exists no coding scheme assigning short code words to many different messages - see example 4.2.  $\diamond$

Since there exists no code which, no matter what  $x^n$  is, always mimics the best code for  $x^n$ , it may make sense to look for the next best thing: does there exist a code which, for all  $x^n \in \mathcal{X}^n$ , is ‘nearly’ (in some sense) as good as  $\hat{L}(x^n)$ ? It turns out that in many cases, the answer is yes: there typically exists codes  $\bar{L}$  such that no matter what  $x^n$  arrives,  $\bar{L}(x^n)$  is not much larger than  $\hat{L}(x^n)$ , which may be viewed as the code that is best ‘with hindsight’ (i.e., after seeing  $x^n$ ). Intuitively, codes which satisfy this property are called universal codes - a more precise definition follows below. The first (but perhaps not foremost) example of a universal code is the two-part code that we have encountered in section 4.3.

#### 4.4.1 Two-part codes as simple universal codes

##### Example 4.10 (Finite $\mathcal{L}$ )

Let  $\mathcal{L}$  be as in example 4.9. We can devise a code  $\bar{L}_{2-p}$  for all  $x^n \in \mathcal{X}^n$  as follows: to encode  $x^n$ , we first encode the  $j \in \{1, \dots, 9\}$  such that  $L_j(x^n) = \min_{L \in \mathcal{L}} L(x^n)$ , using a uniform code. This takes  $\log 9$  bits. We then encode  $x^n$  itself using the code indexed by  $j$ . This takes  $L_j$  bits. Note that in contrast to the naive scheme discussed in example 4.9, the resulting scheme properly defines a prefix code: a decoder can decode  $x^n$  by first decoding  $j$ , and then decoding  $x^n$  using  $L_j$ . Thus, for every possible  $x^n \in \mathcal{X}^n$ , we obtain

$$\bar{L}_{2-p}(x^n) = \min_{L \in \mathcal{L}} L(x^n) + \log 9.$$

For all  $L \in \mathcal{L}$ ,  $\min_{x^n} L(x^n)$  grows linearly in  $n$ :  $\min_{\theta, x^n} \{-\log P(x^n|\theta)\} = -n \log 0.9 \approx 0.15n$ . Unless  $n$  is very small, no matter what  $x^n$  arises, the extra number of bits we need using  $\bar{L}_{2-p}$  compared to  $\hat{L}(x^n)$  is negligible.  $\diamond$

More generally, let  $\mathcal{L} = \{L_1, \dots, L_M\}$  where  $M$  can be arbitrarily large, and the  $L_j$  can be any code length functions we like; they do not necessarily represent Bernoulli distributions any more. By the reasoning of example 4.10, there exists a (two-part) code such that for all  $x^n \in \mathcal{X}^n$ ,

$$\bar{L}_{2-p}(x^n) = \min_{L \in \mathcal{L}} L(x^n) + \log M. \quad (4.11)$$

In most applications  $\min L(x^n)$  grows linearly in  $n$ , and we see from (4.11) that, as soon as  $n$  becomes substantially larger than  $\log M$ , the difference in performance between our universal code and  $\bar{L}(x^n)$  becomes negligible. In general, we do not always want to use a uniform code for the elements in  $\mathcal{L}$ ; note that any arbitrary code on  $\mathcal{L}$  will give us an analogue of (4.11), but with a worst-case overhead larger than  $\log M$  - corresponding to the largest code length of any of the elements in  $\mathcal{L}$ .

#### Example 4.11 (Countable infinite $\mathcal{L}$ )

We can also construct a 2-part code for arbitrary countably infinite sets of codes  $\mathcal{L} = \{L_1, L_2, \dots\}$ : we first encode some  $k$  using our simple code for the integers (example 4.4). With this code we need  $2 \log k + 1$  bits to encode integer  $k$ . We then encode  $x^n$  using the code  $L_k$ .  $\bar{L}_{2-p}$  is now defined as the code we get if, for any  $x^n$ , we encode  $x^n$  using the  $L_k$  minimizing the total two-part description length  $2 \log k + 1 + L_k(x^n)$ .

In contrast to the case of finite  $\mathcal{L}$ , there does not exist a constant  $c$  any more such that for all  $n, x^n \in \mathcal{X}^n$ ,  $\bar{L}_{2-p}(x^n) \leq \inf_{L \in \mathcal{L}} L(x^n) + c$ . Instead we have the following weaker, but still remarkable property: for all  $k$ , all  $n$ , all  $x^n$ ,  $\bar{L}_{2-p}(x^n) \leq L_k(x^n) + 2 \log k + 1$ . Therefore, we also get

$$\bar{L}_{2-p}(x^n) \leq \inf_{L \in \{L_1, \dots, L_k\}} L(x^n) + 2 \log k + 1.$$

For any  $k$ , as  $n$  grows larger, the code  $\bar{L}_{2-p}$  starts to mimic whatever  $L \in \{L_1, \dots, L_k\}$  compresses the data most. However, the larger  $k$ , the larger  $n$  has to be before this happens.  $\diamond$

#### 4.4.2 From universal codes to universal models

Instead of postulating a set of candidate codes  $\mathcal{L}$ , we may equivalently postulate a set  $\mathcal{M}$  of candidate probabilistic sources, such that  $\mathcal{L}$  is the set of codes corresponding to  $\mathcal{M}$ . We already implicitly did this in example 4.9.

The reasoning is now as follows: we think that one of the  $P \in \mathcal{M}$  will assign a high likelihood to the data to be observed. Therefore we would like to design a code that, for all  $x^n$  we might observe, performs essentially as well as the code corresponding to the best-fitting, maximum likelihood (minimum code length)  $P \in \mathcal{M}$  for  $x^n$ . Similarly, we can think of universal codes such as the two-part code in terms of the (possibly defective, see section 4.1 and observation 4.1) distributions corresponding to it. Such distributions corresponding to universal codes are called universal models. The use of mapping universal codes back to distributions is illustrated by the Bayesian universal model which we now introduce.

##### Universal model: twice misleading terminology

The words 'universal' and 'model' are somewhat of a misnomer: first, these codes/models are only 'universal' relative to a restricted 'universe'  $\mathcal{M}$ . Second, the use of the word 'model' will be very confusing to statisticians, who (as we also do in this chapter) call a family of distributions such as  $\mathcal{M}$  a 'model'. But the phrase originates from information theory, where a 'model' often refers to a single distribution rather than a family. Thus, a 'universal model' is a single distribution, representing a statistical 'model'  $\mathcal{M}$ .

**Example 4.12 (Bayesian universal model)**

Let  $\mathcal{M}$  be a finite or countable set of probabilistic sources, parameterized by some parameter set  $\Theta$ . Let  $W$  be a distribution on  $\Theta$ . Adopting terminology from Bayesian statistics,  $W$  is usually called a prior distribution. We can construct a new probabilistic source  $\bar{P}_{Bayes}$  by taking a weighted (according to  $W$ ) average or mixture over the distributions in  $\mathcal{M}$ . That is, we define for all  $n, x^n \in \mathcal{X}$ ,

$$\bar{P}_{Bayes}(x^n) := \sum_{\theta \in \Theta} P(x^n|\theta)W(\theta). \quad (4.12)$$

It is easy to check that  $\bar{P}_{Bayes}$  is a probabilistic source according to our definition. In case  $\Theta$  is continuous, the sum gets replaced by an integral but otherwise nothing changes in the definition. In Bayesian statistics,  $\bar{P}_{Bayes}$  is called the Bayesian marginal likelihood or Bayesian mixture [10]. To see that  $\bar{P}_{Bayes}$  is a universal model, note that for all  $\vartheta \in \Theta$ ,

$$-\log \bar{P}_{Bayes}(x^n) := -\log \sum_{\theta \in \Theta} P(x^n|\theta)W(\theta) \leq -\log P(x^n|\vartheta) + c_\vartheta, \quad (4.13)$$

where the inequality follows because a sum is at least as large as each of its terms, and  $c_\vartheta = -\log W(\vartheta)$  depends on  $\vartheta$  but not on  $n$ . Thus,  $\bar{P}_{Bayes}$  is a universal model or equivalently, the code with lengths  $-\log \bar{P}_{Bayes}$  is a universal code. Note that the derivation in (4.13) only works if  $\Theta$  is finite or countable; the case of continuous  $\Theta$  is treated in section 4.5.  $\diamond$

**Bayes is better than two-part** The Bayesian model is in a sense superior to the two-part code. Namely, in the two-part code we first encode an element of  $\mathcal{M}$  or its parameter set  $\Theta$  using some code  $L_0$ . Such a code must correspond to some ‘prior’ distribution  $W$  on  $\mathcal{M}$  so that the two-part code gives code lengths

$$\bar{L}_{2-p}(x^n) = \min_{\theta \in \Theta} \{-\log P(x^n|\theta) - \log W(\theta)\}, \quad (4.14)$$

where  $W$  depends on the specific code  $L_0$  that was used. Using the Bayes code with prior  $W$ , we get as in (4.13),

$$-\log \bar{P}_{Bayes}(x^n) = -\log \sum_{\theta \in \Theta} P(x^n|\theta)W(\theta) \leq \min_{\theta \in \Theta} \{-\log P(x^n|\theta) - \log W(\theta)\}.$$

The inequality becomes strict whenever  $P(x^n|\theta) > 0$  for more than one value of  $\theta$ . Comparing to (4.14), we see that in general the Bayesian code is preferable over the two-part code: for all  $x^n$  it never assigns code lengths larger than  $\bar{L}_{2-p}(x^n)$ , and in many cases it assigns strictly shorter code lengths for some  $x^n$ . But this raises two important issues: (1) what exactly do we mean by ‘better’ anyway? (2) can we say that ‘some prior distributions are better than others’? These questions are answered below.

**4.4.3 NML as an optimal universal model**

We can measure the performance of universal models relative to a set of candidate sources  $\mathcal{M}$  using the regret:



**Definition 4.1 (Regret)**

Let  $\mathcal{M}$  be a class of probabilistic sources. Let  $\bar{P}$  be a probability distribution on  $\mathcal{X}^n$  ( $\bar{P}$  is not necessarily in  $\mathcal{M}$ ). For given  $x^n$ , the regret,  $\mathcal{R}$ , of  $\bar{P}$  relative to  $\mathcal{M}$  is defined as

$$\mathcal{R} = -\log \bar{P}(x^n) - \min_{P \in \mathcal{M}} \{-\log P(x^n)\}. \quad (4.15)$$

The regret of  $\bar{P}$  relative to  $\mathcal{M}$  for  $x^n$  is the additional number of bits needed to encode  $x^n$  using the code/distribution  $\bar{P}$ , as compared to the number of bits that had been needed if we had used code/distribution in  $\mathcal{M}$  that was optimal ('best-fitting') with hind-sight. For simplicity, from now on we tacitly assume that for all the models  $\mathcal{M}$  we work with, there is a single  $\hat{\theta}(x^n)$  maximizing the likelihood for every  $x^n \in \mathcal{X}^n$ . In that case (4.15) simplifies to

$$\mathcal{R} = -\log \bar{P}(x^n) - \{-\log P(x^n|\hat{\theta}(x^n))\}.$$

We would like to measure the quality of a universal model  $\bar{P}$  in terms of its regret. However,  $\bar{P}$  may have small (even  $< 0$ ) regret for some  $x^n$ , and very large regret for other  $x^n$ . We must somehow find a measure of quality that takes into account all  $x^n \in \mathcal{X}^n$ . We take a worst-case approach, and look for universal models  $\bar{P}$  with small worst-case regret, where the worst-case is over all sequences. Formally, the maximum or worst-case regret of  $\bar{P}$  relative to  $\mathcal{M}$  is defined as

$$\mathcal{R}_{\max}(\bar{P}) := \max_{x^n \in \mathcal{X}^n} \{-\log \bar{P}(x^n) - \{-\log P(x^n|\hat{\theta}(x^n))\}\}.$$

If we use  $\mathcal{R}_{\max}$  as our quality measure, then the 'optimal' universal model relative to  $\mathcal{M}$ , for given sample size  $n$ , is the distribution minimizing

$$\min_{\bar{P}} \mathcal{R}_{\max}(\bar{P}) = \min_{\bar{P}} \max_{x^n \in \mathcal{X}^n} \{-\log \bar{P}(x^n) - \{-\log P(x^n|\hat{\theta}(x^n))\}\}, \quad (4.16)$$

where the minimum is over all defective distributions on  $\mathcal{X}^n$ . The  $\bar{P}$  minimizing (4.16) corresponds to the code minimizing the additional number of bits compared to code in  $\mathcal{M}$  that is best in hindsight in the worst-case over all possible  $x^n$ . It turns out that we can solve for  $\bar{P}$  in (4.16). To this end, we first define the complexity of a given model  $\mathcal{M}$  as

$$\text{COMP}_n(\mathcal{M}) := \log \sum_{x^n \in \mathcal{X}^n} P(x^n|\hat{\theta}(x^n)). \quad (4.17)$$

This quantity plays a fundamental role in refined MDL, section 4.6. To get a first idea of why  $\text{COMP}_n$  is called model complexity, note that the more sequences  $x^n$  with large  $P(x^n|\hat{\theta}(x^n))$ , the larger  $\text{COMP}_n(\mathcal{M})$ . In other words, the more sequences that can be fit well by an element of  $\mathcal{M}$ , the larger  $\mathcal{M}$ 's complexity.

**Proposition 4.1 (Shtarkov [104])**

Suppose that  $\text{COMP}_n(\mathcal{M})$  is finite. Then the minimax regret (4.16) is uniquely achieved for the distribution  $\bar{P}_{nml}$  given by

$$\bar{P}_{nml}(x^n) := \frac{P(x^n|\hat{\theta}(x^n))}{\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}(y^n))}. \quad (4.18)$$

The distribution  $\bar{P}_{nml}$  is known as the Shtarkov distribution or the normalized maximum likelihood (NML) distribution.

*Proof* proposition 4.1: Plug in  $\bar{P}_{nml}$  in (4.16) and notice that for all  $x^n \in \mathcal{X}^n$ ,

$$-\log \bar{P}_{nml}(x^n) - \{-\log P(x^n|\hat{\theta}(x^n))\} = \mathcal{R}_{\max}(\bar{P}_{nml}) = \text{COMP}_n(\mathcal{M}), \quad (4.19)$$

so that  $\bar{P}_{nml}$  achieves the same regret, equal to  $\text{COMP}_n(\mathcal{M})$ , no matter what  $x^n$  actually obtains. Since every distribution  $P$  on  $\mathcal{X}^n$  with  $P \neq \bar{P}_{nml}$  must satisfy  $P(z^n) < \bar{P}_{nml}(z^n)$  for at least one  $z^n \in \mathcal{X}^n$ , it follows that

$$\begin{aligned} \mathcal{R}_{\max}(P) &\geq -\log P(z^n) + \log P(z^n|\hat{\theta}(z^n)) \\ &> -\log \bar{P}_{nml}(z^n) + \log P(z^n|\hat{\theta}(z^n)) = \mathcal{R}_{\max}(\bar{P}_{nml}). \square \end{aligned}$$

$\bar{P}_{nml}$  is quite literally a ‘normalized maximum likelihood’ distribution: it tries to assign to each  $x^n$  the probability of  $x^n$  according to the ML distribution for  $x^n$ . By (4.10), this is not possible: the resulting ‘probabilities’ add to something larger than 1. But we can normalize these ‘probabilities’ by dividing by their  $\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}(y^n))$ , and then we obtain a probability distribution on  $\mathcal{X}^n$  after all.

Whenever  $\mathcal{X}$  is finite, the sum  $\text{COMP}_n(\mathcal{M})$  is finite so that the NML distribution is well-defined. If  $\mathcal{X}$  is countably infinite or continuous-valued, the sum  $\text{COMP}_n(\mathcal{M})$  may be infinite and then the NML distribution may be undefined. In that case, there exists no universal model achieving constant regret as in (4.19). If  $\mathcal{M}$  is parametric, then  $\bar{P}_{nml}$  is typically well-defined as long as we suitably restrict the parameter space. The parametric case forms the basis of ‘refined MDL’ and will be discussed at length in the next section.

#### Summary: Universal codes and models

Let  $\mathcal{M}$  be a family of probabilistic sources. A universal model in an individual sequence sense<sup>†</sup> relative to  $\mathcal{M}$ , in this text simply called a ‘universal model for  $\mathcal{M}$ ’, is a sequence of distributions  $\bar{P}^{(1)}, \bar{P}^{(2)}, \dots$  on  $\mathcal{X}^1, \mathcal{X}^2, \dots$  respectively, such that for all  $P \in \mathcal{M}$  and  $\epsilon > 0$ ,

$$\max_{x^n \in \mathcal{X}^n} \frac{1}{n} \{-\log \bar{P}^{(n)}(x^n) - (-\log P(x^n))\} \leq \epsilon \text{ as } n \rightarrow \infty.$$

Multiplying both sides with  $n$  we see that  $\bar{P}$  is universal if for every  $P \in \mathcal{M}$ , the code length difference  $-\log \bar{P}(x^n) + \log P(x^n)$  increases linearly in  $\epsilon n$ . If  $\mathcal{M}$  is finite, then the two-part, Bayes and NML distributions are universal in a very strong sense: rather than just increasing sublinearly, the code length difference is bounded by a constant. We already discussed two-part, Bayesian and minimax optimal (NML) universal models, but there are several other types. We mention prequential universal models (section 4.5.4), the Kolmogorov universal model, conditionalized two-part codes [97] and Cesaro-average codes [9].

<sup>†</sup> What we call ‘universal model’ in this text is known in the literature as a ‘universal model in the individual sequence sense’ - there also exist universal models in an ‘expected sense’, see section 4.8.1. These lead to slightly different versions of MDL.

#### 4.5 Simple refined MDL and its four interpretations

In section 4.3, we indicated that ‘crude’ MDL needs to be refined. In section 4.4 we introduced universal models. We now show how universal models, in particular the minimax optimal universal model  $\bar{P}_{nml}$ , can be used to define a refined version of MDL model selection. Here we only discuss the simplest case: suppose we are given data  $D = (x_1, \dots, x_n)$  and two models  $\mathcal{M}^{(1)}$  and  $\mathcal{M}^{(2)}$  such that  $\text{COMP}_n(\mathcal{M}^{(1)})$  and  $\text{COMP}_n(\mathcal{M}^{(2)})$  (4.17) are both finite. For example, we could have some binary data and  $\mathcal{M}^{(1)}$  and  $\mathcal{M}^{(2)}$  are the first- and second-order Markov models (example 4.7), both considered possible explanations for the data. We show how to deal with an infinite number of models and/or models with infinite  $\text{COMP}_n$  in section 4.6.

Denote by  $\bar{P}_{nml}(\cdot|\mathcal{M}^{(j)})$  the NML distribution on  $\mathcal{X}^n$  corresponding to model  $\mathcal{M}^{(j)}$ . Refined MDL tells us to pick the model  $\mathcal{M}^{(j)}$  maximizing the normalized maximum likelihood  $\bar{P}_{nml}(D|\mathcal{M}^{(j)})$ , or, by (4.18), equivalently, minimizing

$$-\log \bar{P}_{nml}(D|\mathcal{M}^{(j)}) = -\log P(D|\hat{\theta}^{(j)}(D)) + \text{COMP}_n(\mathcal{M}^{(j)}). \quad (4.20)$$

From a coding theoretic point of view, we associate with each  $\mathcal{M}^{(j)}$  a code with lengths  $\bar{P}_{nml}(\cdot|\mathcal{M}^{(j)})$ , and we pick the model minimizing the code length of the data. The code length  $-\log \bar{P}_{nml}(D|\mathcal{M}^{(j)})$  has been called the stochastic complexity of the data  $D$  relative to model  $\mathcal{M}^{(j)}$  [92], whereas  $\text{COMP}_n(\mathcal{M}^{(j)})$  is called the parametric complexity or model cost of  $\mathcal{M}^{(j)}$  (in this chapter we simply call it ‘complexity’). We have already indicated in the previous section that  $\text{COMP}_n(\mathcal{M}^{(j)})$  measures something like the ‘complexity’ of model  $\mathcal{M}^{(j)}$ . On the other hand,  $-\log P(D|\hat{\theta}^{(j)}(D))$  is minus the maximized log-likelihood of the data, so it measures something like (minus) fit or error - in the linear regression case, it can be directly related to the mean squared error, section 4.8. Thus, (4.20) embodies a trade-off between lack of fit (measured by minus log-likelihood) and complexity (measured by  $\text{COMP}_n(\mathcal{M}^{(j)})$ ). The confidence in the decision is given by the code length difference

$$\left| -\log \bar{P}_{nml}(D|\mathcal{M}^{(1)}) - (-\log \bar{P}_{nml}(D|\mathcal{M}^{(2)})) \right|.$$

In general,  $-\log \bar{P}_{nml}(D|\mathcal{M})$  can only be evaluated numerically - the only exception is when  $\mathcal{M}$  is the Gaussian family, example 4.18. In many cases even numerical evaluation is computationally problematic. But the re-interpretations of  $\bar{P}_{nml}$  we provide below also indicate that in many cases,  $-\log \bar{P}_{nml}(D|\mathcal{M})$  is relatively easy to approximate.

##### Example 4.13 (Refined MDL and GLRT)

Generalized likelihood ratio testing [15] tells us to pick the  $\mathcal{M}^{(j)}$  maximizing  $\log P(D|\hat{\theta}^{(j)}(D)) + c$  where  $c$  is determined by the desired type-I and type-II errors. In practice one often applies a naive variation\*, simply picking the model  $\mathcal{M}^{(j)}$  maximizing  $\log P(D|\hat{\theta}^{(j)}(D))$ . This amounts to ignoring the complexity terms  $\text{COMP}_n(\mathcal{M}^{(j)})$  in (4.20): MDL tries to avoid overfitting by picking the model maximizing the normalized rather than the ordinary likelihood. The more distributions in  $\mathcal{M}$  that fit the data well, the larger the normalization term.  $\diamond$

\* To be fair, we should add that this naive version of GLRT is introduced here for educational purposes only. It is not recommended by any serious statistician!

The hope is that the normalization term  $\text{COMP}_n(\mathcal{M}^{(j)})$  strikes the right balance between complexity and fit. Whether it really does this, depends on whether  $\text{COMP}_n$  is a ‘good’ measure of complexity. In the remainder of this section we shall argue that it is, by giving four different interpretations of  $\text{COMP}_n$  and of the resulting trade-off (4.20):

1. Compression interpretation.
2. Counting interpretation.
3. Bayesian interpretation.
4. Prequential (predictive) interpretation.

#### 4.5.1 Compression interpretation

Rissanen’s original goal was to select the model that detects the most regularity in the data; he identified this with the ‘model that allows for the most compression of data  $x^n$ ’. To make this precise, a code is associated with each model. The NML code with lengths  $-\log \bar{P}_{nml}(\cdot|\mathcal{M}^{(j)})$  seems to be a very reasonable choice for such a code because of the following two properties:

1. The better the best-fitting distribution in  $\mathcal{M}^{(j)}$  fits the data, the shorter the code length  $-\log \bar{P}_{nml}(D|\mathcal{M}^{(j)})$ .
2. No distribution in  $\mathcal{M}^{(j)}$  is given a prior preference over any other distribution, since the regret of  $\bar{P}_{nml}(\cdot|\mathcal{M}^{(j)})$  is the same for all  $D \in \mathcal{X}^n$  (4.19).  $\bar{P}_{nml}$  is the only complete prefix code with this property, which may be restated as:  $\bar{P}_{nml}$  treats all distributions within each  $\mathcal{M}^{(j)}$  on the same footing!

Therefore, if one is willing to accept the basic ideas underlying MDL as first principles, then the use of NML in model selection is now justified to some extent. Below we give additional justifications that are not directly based on data compression; but we first provide some further interpretation of  $-\log \bar{P}_{nml}$ .

**Compression and separating structure from noise** We present the following ideas in an imprecise fashion - Rissanen and Tabus [99] recently showed how to make them precise. The stochastic complexity of data  $D$  relative to  $\mathcal{M}$ , given by (4.20) can be interpreted as the amount of information in the data relative to  $\mathcal{M}$ , measured in bits. Although a one-part code length, it still consists of two terms: a term  $\text{COMP}_n(\mathcal{M})$  measuring the amount of structure or meaningful information in the data (as ‘seen through  $\mathcal{M}$ ’), and a term  $-\log P(D|\hat{\theta}(D))$  measuring the amount of noise or accidental information in the data. To see that this second term measures noise, consider the regression example, example 3.2, again. As will be seen in section 4.8, in that case  $-\log P(D|\hat{\theta}(D))$  becomes equal to a linear function of the mean squared error of the best-fitting polynomial in the set of  $k$ -th degree polynomials. To see that the first term measures structure, we reinterpret it below as the number of bits needed to specify a ‘distinguishable’ distribution in  $\mathcal{M}$ , using a uniform code on all ‘distinguishable’ distributions.

#### 4.5.2 Counting interpretation

The parametric complexity can be interpreted as measuring (the log of) the number of distinguishable distributions in the model. Intuitively, the more distributions a model contains, the more patterns it can fit well, so the larger the risk of overfitting. However,

if two distributions are very ‘close’ in the sense that they assign high likelihood to the same patterns, they do not contribute so much to the complexity of the overall model. It seems that we should measure complexity of a model in terms of the number of distributions it contains that are ‘essentially different’ (distinguishable), and we now show that  $\text{COMP}_n$  measures something like this. Consider a finite model  $\mathcal{M}$  with parameter set  $\Theta = \{\theta_1, \dots, \theta_M\}$ . Note that

$$\begin{aligned} \sum_{x^n \in \mathcal{X}^n} P(x^n | \hat{\theta}(x^n)) &= \sum_{j=1 \dots M} \sum_{\substack{x^n \\ \hat{\theta}(x^n) = \theta_j}} P(x^n | \theta_j) = \sum_{j=1 \dots M} \left( 1 - \sum_{\substack{x^n \\ \hat{\theta}(x^n) \neq \theta_j}} P(x^n | \theta_j) \right) \\ &= M - \sum_j P(\hat{\theta}(x^n) \neq \theta_j | \theta_j). \end{aligned}$$

We may think of  $P(\hat{\theta}(x^n) \neq \theta_j | \theta_j)$  as the probability, according to  $\theta_j$ , that the data look as if they come from some  $\theta \neq \theta_j$ . Thus, it is the probability that  $\theta_j$  is mistaken for another distribution in  $\Theta$ . Therefore, for finite  $\mathcal{M}$ , Rissanen’s model complexity is the logarithm of the number of distributions minus the summed probability that some  $\theta_j$  is ‘mistaken’ for some  $\theta \neq \theta_j$ . Now suppose  $\mathcal{M}$  is i.i.d. By the law of large numbers [30], we immediately see that the ‘sum of mistake probabilities’  $\sum_j P(\hat{\theta}(x^n) \neq \theta_j | \theta_j)$  tends to 0 as  $n$  grows. It follows that for large  $n$ , the model complexity converges to  $\log M$ . For large  $n$ , the distributions in  $\mathcal{M}$  are ‘perfectly distinguishable’ (the probability that a sample coming from one is more representative of another is negligible), and then the parametric complexity  $\text{COMP}_n(\mathcal{M})$  of  $\mathcal{M}$  is simply the log of the number of distributions in  $\mathcal{M}$ .

#### Example 4.14 (NML vs. two-part codes)

Incidentally, this shows that for finite i.i.d.  $\mathcal{M}$ , the two-part code with uniform prior  $W$  on  $\mathcal{M}$  is asymptotically minimax optimal: for all  $n$ , the regret of the 2-part code is  $\log M$  (4.11), whereas we just showed that for  $n \rightarrow \infty$ ,  $\mathcal{R}(\bar{P}_{nml}) = \text{COMP}_n(\mathcal{M}) \rightarrow \log M$ . However, for small  $n$ , some distributions in  $\mathcal{M}$  may be mistaken for one another; the number of distinguishable distributions in  $\mathcal{M}$  is then smaller than the actual number of distributions, and this is reflected in  $\text{COMP}_n(\mathcal{M})$  being (sometimes much) smaller than  $\log M$ .  $\diamond$

**Asymptotic expansion of  $\bar{P}_{nml}$  and  $\text{COMP}_n$**  Let  $\mathcal{M}$  be a  $k$ -dimensional parametric model. Under regularity conditions on  $\mathcal{M}$  and the parameterization  $\Theta \rightarrow \mathcal{M}$ , to be detailed below, we obtain the following asymptotic expansion ( $n \rightarrow \infty$ ) of  $\text{COMP}_n$  [95], [109], [110], [111]:

$$\text{COMP}_n(\mathcal{M}) = \frac{k}{2} \log \frac{n}{2\pi} + \log \int_{\theta \in \Theta} \sqrt{|I(\theta)|} d\theta + \mathcal{O}(1). \quad (4.21)$$

Here  $k$  is the number of parameters (degrees of freedom) in model  $\mathcal{M}$ ,  $n$  is the sample size, and  $\mathcal{O}(1) \rightarrow 0$  as  $n \rightarrow \infty$ .  $|I(\theta)|$  is the determinant of the  $k \times k$  Fisher information

matrix<sup>†</sup>  $I$  evaluated at  $\theta$ . In case  $\mathcal{M}$  is an i.i.d. model,  $I$  is given by

$$I_{ij}(\theta^*) := E_{\theta^*} \left\{ -\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log P(X|\theta) \right\}_{\theta=\theta^*}.$$

This is generalized to non-i.i.d. models as follows:

$$I_{ij}(\theta^*) := \lim_{n \rightarrow \infty} \frac{1}{n} E_{\theta^*} \left\{ -\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log P(X^n|\theta) \right\}_{\theta=\theta^*}.$$

Equation (4.21) only holds if the model  $\mathcal{M}$ , its parameterization  $\Theta$  and the sequence  $x_1, x_2, \dots$  all satisfy certain conditions. Specifically, we require:

1.  $\text{COMP}_n(\mathcal{M}) < \infty$  and  $\int \sqrt{|I(\theta)|} d\theta < \infty$ .
2.  $\hat{\theta}(x^n)$  does not come arbitrarily close to the boundary of  $\Theta$ : for some  $\epsilon > 0$ , for all large  $n$ ,  $\hat{\theta}(x^n)$  remains farther than  $\epsilon$  away from the boundary of  $\Theta$ .
3.  $\mathcal{M}$  and  $\Theta$  satisfy certain further conditions. A simple sufficient condition is that  $\mathcal{M}$  be an exponential family [15]. Roughly, this is a family that can be parameterized so that for all  $x$ ,  $P(x|\beta) = \exp(\beta t(x)) f(x) g(\beta)$ , where  $t : \mathcal{X} \rightarrow \mathbb{R}$  is a function of  $X$ . The Bernoulli model is an exponential family, as can be seen by setting  $\beta := \ln(1 - \theta) - \ln \theta$  and  $t(x) = x$ . Also the multinomial, Gaussian, Poisson, Gamma, exponential, Zipf and many other models are exponential families; but, for example, mixture models are not.

More general conditions are given by Takeuchi and Barron [109], [110], [111]. Essentially, if  $\mathcal{M}$  behaves ‘asymptotically’ like an exponential family, then (4.21) still holds. For example, (4.21) holds for the Markov models and for AR and ARMA processes.

#### Example 4.15 (Complexity of Bernoulli model)

The Bernoulli model  $\mathcal{B}^{(0)}$  can be parameterized in a 1-1 way by the unit interval (example 4.7). Thus,  $k = 1$ . An easy calculation shows that the Fisher information is given by  $\theta(1 - \theta)$ . Plugging this into (4.21) and calculating  $\int \sqrt{|\theta(1 - \theta)|} d\theta$  gives

$$\text{COMP}_n(\mathcal{B}^{(0)}) = \frac{1}{2} \log n + \frac{1}{2} \log \frac{\pi}{2} - 3 + \mathcal{O}(1) = \frac{1}{2} \log n - 2.674251935 + \mathcal{O}(1).$$

Computing the integral of the Fisher determinant is not easy in general. Hanson and Fu [53] compute it for several practically relevant models.  $\diamond$

Whereas for finite  $\mathcal{M}$ ,  $\text{COMP}_n(\mathcal{M})$  remains finite, for parametric models it generally grows logarithmically in  $n$ . Since typically  $-\log P(x^n|\hat{\theta}(x^n))$  grows linearly in  $n$ , it is still the case that for fixed dimensionality  $k$  (i.e. for a fixed  $\mathcal{M}$  that is  $k$ -dimensional) and large  $n$ , the part of the code length  $-\log \bar{P}_{nml}(x^n|\mathcal{M})$  due to the complexity of  $\mathcal{M}$  is very small compared to the part needed to encode data  $x^n$  with  $\hat{\theta}(x^n)$ . The term  $\int_{\Theta} \sqrt{|I(\theta)|} d\theta$  may be interpreted as the contribution of the functional form of  $\mathcal{M}$  to the model complexity [5]. It does not grow with  $n$  so that, when selecting between two models, it becomes irrelevant and can be ignored for very large  $n$ . But for small  $n$ , it can be important, as can be seen from example 3.4, Fechner’s and Stevens’ model. Both

<sup>†</sup> The standard definition of Fisher information [63] is in terms of first derivatives of the log-likelihood; for most parametric models of interest, the present definition coincides with the standard one.

models have two parameters, yet the  $\int_{\Theta} \sqrt{|I(\theta)|} d\theta$ -term is much larger for Fechner's than for Stevens' model. In the experiments in [80], the parameter set was restricted to  $0 < a < 1$ ,  $0 < b < 3$  for Stevens' model and  $0 < a < 1$ ,  $0 < b < 1$  for Fechner's model. The variance of the error  $Z$  was set to 1 in both models. With these values, the difference in  $\int_{\Theta} \sqrt{|I(\theta)|} d\theta$  is 3.804, which is non-negligible for small samples. Thus, Stevens' model contains more distinguishable distributions than Fechner's, and is better able to capture random noise in the data - as Townsend [113] already speculated almost 30 years ago. Experiments suggest that for regression models such as Stevens' and Fechner's, as well as for Markov models and general exponential families, the approximation (4.21) is reasonably accurate already for small samples. But this is certainly not true for general models:

**The asymptotic expansion of  $\text{COMP}_n$  should be used with care!**

Equation (4.21) does not hold for all parametric models; and for some models for which it does hold, the  $\mathcal{O}(1)$  term may only converge to 0 only for quite large sample sizes. In [32], [34] it is shown that the approximation (4.21) is, in general, only valid if  $k$  is much smaller than  $n$ .

**Two-part codes and  $\text{COMP}_n(\mathcal{M})$**  We now have a clear guiding principle (minimax regret) which we can use to construct 'optimal' two-part codes, that achieve the minimax regret among all two-part codes. How do such optimal two-part codes compare to the NML code length? Let  $\mathcal{M}$  be a  $k$ -dimensional model. By slightly adjusting the arguments of [8], one can show that, under regularity conditions, the minimax optimal two-part code  $\bar{P}_{2-p}$  achieves regret

$$\begin{aligned} \mathcal{R} &= -\log \bar{P}_{2-p}(x^n | \mathcal{M}) + \log P(x^n | \hat{\theta}(x^n)) \\ &= \frac{k}{2} \log \frac{n}{2\pi} + \log \int_{\theta \in \Theta} \sqrt{|I(\theta)|} d\theta + f(k) + \mathcal{O}(1), \end{aligned}$$

where  $f : \mathbb{N} \rightarrow \mathbb{R}$  is a bounded positive function satisfying  $\lim_{k \rightarrow \infty} f(k) = 0$ . Thus, for large  $k$ , optimally designed two-part codes are about as good as NML. The problem with two-part code MDL is that in practice, people often use much cruder codes with much larger minimax regret.

### 4.5.3 Bayesian interpretation

The Bayesian method of statistical inference provides several alternative approaches to model selection. The most popular of these is based on Bayes factors [63]. The Bayes factor method is very closely related to the refined MDL approach. Assuming uniform priors on models  $\mathcal{M}^{(1)}$  and  $\mathcal{M}^{(2)}$ , it tells us to select the model with largest marginal likelihood  $\bar{P}_{\text{Bayes}}(x^n | \mathcal{M}^{(j)})$ .  $\bar{P}_{\text{Bayes}}$  is as in (4.12), with the sum replaced by an integral:

$$\bar{P}_{\text{Bayes}}(x^n | \mathcal{M}^{(j)}) = \int P(x^n | \theta) w^{(j)}(\theta) d\theta, \quad (4.22)$$

where  $w^{(j)}$  is the density of the prior distribution on  $\mathcal{M}^{(j)}$ .

**$\mathcal{M}$  is an exponential family** Let now  $\bar{P}_{\text{Bayes}} = \bar{P}_{\text{Bayes}}(\cdot | \mathcal{M})$  for some fixed model  $\mathcal{M}$ . Under regularity conditions on  $\mathcal{M}$ , we can perform a Laplace approximation of

the integral in (4.12). For the special case that  $\mathcal{M}$  is an exponential family, we obtain the following expression for the regret [59], [100], [63], [4]:

$$\begin{aligned}\mathcal{R} &= -\log \bar{P}_{Bayes}(x^n) - (-\log P(x^n|\hat{\theta}(x^n))) \\ &= \frac{k}{2} \log \frac{n}{2\pi} - \log w(\hat{\theta}) + \log \sqrt{|I(\hat{\theta})|} + \mathcal{O}(1).\end{aligned}\quad (4.23)$$

Let us compare this with (4.21). Under the regularity conditions needed for (4.21), the quantity on the right hand side of (4.23) is within  $\mathcal{O}(1)$  of  $\text{COMP}_n(\mathcal{M})$ . Thus, the code length achieved with  $\bar{P}_{Bayes}$  is the minimax optimum  $-\log \bar{P}_{nml}(x^n)$ , apart from a constant. Since  $-\log P(x^n|\hat{\theta}(x^n))$  increases linearly in  $n$ , this means that if we compare two models  $\mathcal{M}^{(1)}$  and  $\mathcal{M}^{(2)}$ , then for large enough  $n$ , Bayes and refined MDL select the same model. If we equip the Bayesian universal model with a special prior known as the Jeffreys-Bernardo prior [59], [10],

$$w_{Jeffreys}(\theta) = \frac{\sqrt{|I(\theta)|}}{\int_{\theta \in \Theta} \sqrt{|I(\theta)|} d\theta}, \quad (4.24)$$

then Bayes and refined NML become even more closely related: plugging in (4.24) into (4.23), we find that the right-hand side of (4.23) now simply coincides with (4.21). A concrete example of Jeffreys' prior is given in example 4.19. Jeffreys introduced his prior as a 'least informative prior', to be used when no useful prior knowledge about the parameters is available [59]. As one may expect from such a prior, it is invariant under continuous 1-to-1 reparameterizations of the parameter space. The present analysis shows that, when  $\mathcal{M}$  is an exponential family, then it also leads to asymptotically minimax code length regret: for large  $n$ , refined NML model selection becomes indistinguishable from Bayes factor model selection with Jeffreys' prior.

**$\mathcal{M}$  is not an exponential family** Under weak conditions on  $\mathcal{M}$ ,  $\Theta$  and the sequence  $x^n$ , we get the following generalization of (4.23):

$$\begin{aligned}-\log \bar{P}_{Bayes}(x^n|\mathcal{M}) &= \\ &= -\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2} \log \frac{n}{2\pi} - \log w(\hat{\theta}) + \log \sqrt{|\hat{I}(x^n)|} + \mathcal{O}(1).\end{aligned}\quad (4.25)$$

Here  $\hat{I}(x^n)$  is the so-called observed information, sometimes also called observed Fisher information; see [63] for a definition. If  $\mathcal{M}$  is an exponential family, then the observed Fisher information at  $x^n$  coincides with the Fisher information at  $\hat{\theta}(x^n)$ , leading to (4.23). If  $\mathcal{M}$  is not exponential, then provided that the data are distributed according to one of the distributions in  $\mathcal{M}$ , the observed Fisher information still converges with probability 1 to the expected Fisher information. If  $\mathcal{M}$  is neither exponential, nor are the data actually generated by a distribution in  $\mathcal{M}$ , then there may be  $\mathcal{O}(1)$ -discrepancies between  $-\log \bar{P}_{nml}$  and  $-\log \bar{P}_{Bayes}$  even for large  $n$ .

#### 4.5.4 Prequential interpretation

**Distributions as prediction strategies** Let  $P$  be a distribution on  $\mathcal{X}^n$ . Applying the definition of conditional probability, we can write for every  $x^n$ :

$$P(x^n) = \prod_{i=1}^n \frac{P(x^i)}{P(x^{i-1})} = \prod_{i=1}^n P(x_i|x^{i-1}), \quad (4.26)$$



so that also

$$-\log P(x^n) = \sum_{i=1}^n -\log P(x_i|x^{i-1}). \quad (4.27)$$

Let us abbreviate  $P(X_i = \cdot | X^{i-1} = x^{i-1})$  to  $P(X_i|x^{i-1})$ . Note that  $P(X_i|x^{i-1})$  (capital  $X_i$ ) is the distribution (not a single number) of  $X_i$  given  $x^{i-1}$ ;  $P(x_i|x^{i-1})$  (lower case  $x_i$ ) is the probability (a single number) of actual outcome  $x_i$  given  $x^{i-1}$ . We can think of  $-\log P(x_i|x^{i-1})$  as the loss incurred when predicting  $X_i$  based on the conditional distribution  $P(X_i|x^{i-1})$ , and the actual outcome turned out to be  $x_i$ . Here ‘loss’ is measured using the so-called logarithmic score, also known simply as ‘log loss’. Note that the more likely  $x$  is judged to be, the smaller the loss incurred when  $x$  actually is obtained. The log loss has a natural interpretation in terms of sequential gambling [20], but its main interpretation is still in terms of coding: by (4.27), the code length needed to encode  $x^n$  based on distribution  $P$  is just the accumulated log loss incurred when  $P$  is used to sequentially predict the  $i$ -th outcome based on the past  $(i-1)$ -st outcomes.

Equation (4.26) gives a fundamental re-interpretation of probability distributions as prediction strategies, mapping each individual sequence of past observations  $x_1, \dots, x_{i-1}$  to a probabilistic prediction of the next outcome  $P(X_i|x^{i-1})$ . Conversely, (4.26) also shows that every probabilistic prediction strategy for sequential prediction of  $n$  outcomes may be thought of as a probability distribution on  $\mathcal{X}^n$ : a strategy is identified with a function mapping all potential initial segments  $x^{i-1}$  to the prediction that is made for the next outcome  $X_i$ , after having seen  $x^{i-1}$ . Thus, it is a function  $S : \bigcup_{0 \leq i < n} \mathcal{X}^i \rightarrow \mathcal{P}_{\mathcal{X}}$ , where  $\mathcal{P}_{\mathcal{X}}$  is the set of distributions on  $\mathcal{X}$ . We can now define, for each  $i < n$ , all  $x^i \in \mathcal{X}^i$ ,  $P(X_i|x^{i-1}) := S(x^{i-1})$ . We can turn these partial distributions into a distribution on  $\mathcal{X}^n$  by sequentially plugging them into (4.26).

**Log loss for universal models** Let  $\mathcal{M}$  be some parametric model and let  $\bar{P}$  be some universal model/code relative to  $\mathcal{M}$ . What do the individual predictions  $\bar{P}(X_i|x^{i-1})$  look like? Readers familiar with Bayesian statistics will realize that for i.i.d. models, the Bayesian predictive distribution  $\bar{P}_{Bayes}(X_i|x^{i-1})$  converges to the ML distribution  $P(\cdot|\hat{\theta}(x^{i-1}))$ ; example 4.17 provides a concrete case. It seems reasonable to assume that something similar holds not just for  $\bar{P}_{Bayes}$  but for universal models in general. This in turn suggests that we may approximate the conditional distributions  $\bar{P}(X_i|x^{i-1})$  of any ‘good’ universal model by the maximum likelihood predictions  $P(\cdot|\hat{\theta}(x^{i-1}))$ . Indeed, we can recursively define the ‘maximum likelihood plug-in’ distribution  $\bar{P}_{plug-in}$  by setting, for  $i = 1$  to  $n$ ,

$$\bar{P}_{plug-in}(X_i = \cdot | x^{i-1}) := P(X = \cdot | \hat{\theta}(x^{i-1})). \quad (4.28)$$

Then, we define

$$-\log \bar{P}_{plug-in}(x^n) := \sum_{i=1}^n -\log P(x_i|\hat{\theta}(x^{i-1})). \quad (4.29)$$

Indeed, it turns out that under regularity conditions on  $\mathcal{M}$  and  $x^n$ ,

$$-\log \bar{P}_{plug-in}(x^n) = -\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2} \log n + \mathcal{O}(1). \quad (4.30)$$

This shows that  $\bar{P}_{plug-in}$  acts as a universal model relative to  $\mathcal{M}$ , its performance being within a constant of the minimax optimal  $\bar{P}_{nml}$ . The construction of  $\bar{P}_{plug-in}$  can be easily extended to non-i.i.d. models, and then, under regularity conditions, (4.30) still holds; we omit the details.

We note that all general proofs of (4.30) that we are aware of show that (4.30) holds with probability 1 or in expectation for sequences generated by some distribution in  $\mathcal{M}$  [90], [91], [93]. Note that the expressions (4.21) and (4.25) for the regret of  $\bar{P}_{nml}$  and  $\bar{P}_{Bayes}$  hold for a much wider class of sequences; they also hold with probability 1 for i.i.d. sequences generated by sufficiently regular distributions outside  $\mathcal{M}$ . Not much is known about the regret obtained by  $\bar{P}_{plug-in}$  for such sequences, except for some special cases such as  $\mathcal{M}$  being the Gaussian model.

In general, there is no need to use the ML estimator  $\hat{\theta}(x^{i-1})$  in the definition (4.28). Instead, we may try some other estimator which asymptotically converges to the ML estimator - it turns out that some estimators considerably outperform the ML estimator in the sense that (4.29) becomes a much better approximation of  $-\log \bar{P}_{nml}$ , see example 4.17. Irrespective of whether we use the ML estimator or something else, we call model selection based on (4.29) the prequential form of MDL in honor of A.P. Dawid's 'prequential analysis', section 4.8. It is also known as 'predictive MDL'. The validity of (4.30) was discovered independently by Rissanen [90] and Dawid [22].

The prequential view gives us a fourth interpretation of refined MDL model selection: given models  $\mathcal{M}^{(1)}$  and  $\mathcal{M}^{(2)}$ , MDL tells us to pick the model that minimizes the accumulated prediction error resulting from sequentially predicting future outcomes given all the past outcomes.

#### Example 4.16 (GLRT and prequential model selection)

How does this differ from the naive version of the generalized likelihood ratio test (GLRT) that we introduced in example 4.15? In GLRT, we associate with each model the log-likelihood (minus log loss) that can be obtained by the ML estimator. This is the predictor within the model that minimizes log loss with hindsight, after having seen the data. In contrast, prequential model selection associates with each model the log-likelihood (minus log loss) that can be obtained by using a sequence of ML estimators  $\hat{\theta}(x^{i-1})$  to predict data  $x_i$ . Crucially, the data on which ML estimators are evaluated has not been used in constructing the ML estimators themselves. This makes the prediction scheme 'honest' (different data are used for training and testing) and explains why it automatically protects us against overfitting.  $\diamond$

#### Example 4.17 (Laplace and Jeffreys)

Consider the prequential distribution for the Bernoulli model, example 4.7, defined as in (4.28). We show that if we take  $\hat{\theta}$  in (4.28) equal to the ML estimator  $\frac{n_{[1]}}{n}$ , then the resulting  $\bar{P}_{plug-in}$  is not a universal model; but a slight modification of the ML estimator makes  $\bar{P}_{plug-in}$  a very good universal model. Suppose that  $n \geq 3$  and  $(x_1, x_2, x_3) = (0, 0, 1)$  - a not-so-unlikely initial segment according to most  $\theta$ . Then  $\bar{P}_{plug-in}(X_3 = 1 | x_1, x_2) = P(X = 1 | \hat{\theta}(x_1, x_2)) = 0$ , so that by (4.29), we get

$$-\log \bar{P}_{plug-in}(x^n) \geq -\log \bar{P}_{plug-in}(x_3 | x_1, x_2) = \infty,$$

hence  $\bar{P}_{plug-in}$  is not universal. Now let us consider the modified ML estimator

$$\hat{\theta}_\lambda(x^n) := \frac{n_{[1]} + \lambda}{n + 2\lambda}. \quad (4.31)$$

If we take  $\lambda = 0$ , we get the ordinary ML estimator. If we take  $\lambda = 1$ , then an exercise involving beta-integrals shows that, for all  $i, x^i$ ,  $P(X_i|\hat{\theta}_1(x^{i-1})) = \bar{P}_{Bayes}(X_i|x^{i-1})$ , where  $\bar{P}_{Bayes}$  is defined relative to the uniform prior  $w(\theta) \equiv 1$ . Thus  $\hat{\theta}_1(x^{i-1})$  corresponds to the Bayesian predictive distribution for the uniform prior. This prediction rule was advocated by the great probabilist P.S. de Laplace, co-originator of Bayesian statistics. It may be interpreted as ML estimation based on an extended sample, containing some ‘virtual’ data: an extra 0 and an extra 1.

Even better, a similar calculation shows that if we take  $\lambda = 2$ , the resulting estimator is equal to  $\bar{P}_{Bayes}(X_i|x^{i-1})$  defined relative to Jeffreys’ prior. Asymptotically,  $\bar{P}_{Bayes}$  with Jeffreys’ prior achieves the same code lengths as  $\bar{P}_{nml}$  (section 4.5.3). It follows that  $\bar{P}_{plug-in}$  with the slightly modified ML estimator is asymptotically indistinguishable from the optimal universal model  $\bar{P}_{nml}$ !

For more general models  $\mathcal{M}$ , such simple modifications of the ML estimator usually do not correspond to a Bayesian predictive distribution; for example, if  $\mathcal{M}$  is not convex (closed under taking mixtures) then a point estimator (an element of  $\mathcal{M}$ ) typically does not correspond to the Bayesian predictive distribution (a mixture of elements of  $\mathcal{M}$ ). Nevertheless, modifying the ML estimator by adding some virtual data  $y_1, \dots, y_m$  and replacing  $P(X_i|\hat{\theta}(x^{i-1}))$  by  $P(X_i|\hat{\theta}(x^{i-1}, y^m))$  in the definition (4.28) may still lead to good universal models. This is of great practical importance, since, using (4.29),  $-\log \bar{P}_{plug-in}(x^n)$  is often much easier to compute than  $-\log \bar{P}_{Bayes}(x^n)$ .  $\diamond$

**Summary** We introduced the refined MDL principle for model selection in a restricted setting. Refined MDL amounts to selecting the model under which the data achieve the smallest stochastic complexity, which is the code length according to the minimax optimal universal model. We gave an asymptotic expansion of stochastic and parametric complexity, and interpreted these concepts in four different ways.

## 4.6 General refined MDL: gluing it all together

In the previous section we introduced a ‘refined’ MDL principle based on minimax regret. Unfortunately, this principle can be applied only in very restricted settings. We now show how to extend refined MDL, leading to a general MDL principle, applicable to a wide variety of model selection problems. In doing so we glue all our previous insights (including ‘crude MDL’) together, thereby uncovering a single general, underlying principle, formulated in observation 4.4. Therefore, if one understands the material in this section, then one understands the minimum description length principle.

First, in section 4.6.1, we show how to compare infinitely many models. Then, section 4.6.2 shows how to proceed for models  $\mathcal{M}$  for which the parametric complexity is undefined. Remarkably, a single, general idea resides behind our solution of both problems, and this leads us to formulate, in section 4.6.3, a single, general refined MDL principle.

#### 4.6.1 Model selection with infinitely many models

Suppose we want to compare more than two models for the same data. If the number to be compared is finite, we can proceed as before and pick the model  $\mathcal{M}^{(k)}$  with smallest  $-\log \bar{P}_{nml}(x^n|\mathcal{M}^{(k)})$ . If the number of models is infinite, we have to be more careful. Say we compare models  $\mathcal{M}^{(1)}, \mathcal{M}^{(2)}, \dots$  for data  $x^n$ . We may be tempted to pick the model minimizing  $-\log \bar{P}_{nml}(x^n|\mathcal{M}^{(k)})$  over all  $k \in \{1, 2, \dots\}$ , but in some cases this gives unintended results. To illustrate, consider the extreme case that every  $\mathcal{M}^{(k)}$  contains just one distribution. For example, let  $\mathcal{M}^{(1)} = \{P_1\}$ ,  $\mathcal{M}^{(2)} = \{P_2\}$ ,  $\dots$  where  $\{P_1, P_2, \dots\}$  is the set of all Markov chains with rational-valued parameters. In that case,  $\text{COMP}_n(\mathcal{M}^{(k)}) = 0$  for all  $k$ , and we would always select the maximum likelihood Markov chain that assigns probability 1 to data  $x^n$ . Typically this will be a chain of very high order, severely overfitting the data. This cannot be right! A better idea is to pick the model minimizing

$$-\log \bar{P}_{nml}(x^n|\mathcal{M}^{(k)}) + L(k), \quad (4.32)$$

where  $L$  is the code length function of some code for encoding model indices  $k$ . We would typically choose the standard prior for the integers,  $L(k) = 2 \log k + 1$ , example 4.4. By using (4.32) we avoid the overfitting problem mentioned above: if  $\mathcal{M}^{(1)} = \{P_1\}$ ,  $\mathcal{M}^{(2)} = \{P_2\}$ ,  $\dots$  where  $P_1, P_2, \dots$  is a list of all the rational-parameter Markov chains, (4.32) would reduce to two-part code MDL (section 4.3) which is asymptotically consistent. On the other hand, if  $\mathcal{M}^{(k)}$  represents the set of  $k$ -th order Markov chains, the term  $L(k)$  is typically negligible compared to  $\text{COMP}_n(\mathcal{M}^{(k)})$ , the complexity term associated with  $\mathcal{M}^{(k)}$  that is hidden in  $-\log \bar{P}_{nml}(\mathcal{M}^{(k)})$ . Thus, the complexity of  $\mathcal{M}^{(k)}$  comes from the fact that for large  $k$ ,  $\mathcal{M}^{(k)}$  contains many distinguishable distributions; not from the much smaller term  $L(k) \approx 2 \log k$ .

To make our previous approach for a finite set of models compatible with (4.32), we can reinterpret it as follows: we assign uniform code lengths (a uniform prior) to the  $\mathcal{M}^{(1)}, \dots, \mathcal{M}^{(M)}$  under consideration, so that for  $k = 1, \dots, M$ ,  $L(k) = \log M$ . We then pick the model minimizing (4.32). Since  $L(k)$  is constant over  $k$ , it plays no role in the minimization and can be dropped from the equation, so that our procedure reduces to our original refined MDL model selection method. We shall henceforth assume that we always encode the model index, either implicitly (if the number of models is finite) or explicitly. The general principle behind this is explained in section 4.6.3.

#### 4.6.2 The infinity problem

For some of the most commonly used models, the parametric complexity  $\text{COMP}(\mathcal{M})$  is undefined. A prime example is the Gaussian location model, which we discuss below. As we will see, we can ‘repair’ the situation using the same general idea as in the previous subsection.

##### Example 4.18 (Parametric complexity of the normal distributions)

Let  $\mathcal{M}$  be the family of normal distributions with fixed variance  $\sigma^2$  and varying mean  $\mu$ , identified by their densities

$$P(x|\mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

extended to sequences  $x_1, \dots, x_n$  by taking product densities. As is well-known [15], the ML estimator  $\hat{\mu}(x^n)$  is equal to the sample mean:  $\hat{\mu}(x^n) = n^{-1} \sum_{i=1}^n x_i$ . An easy calculation shows that

$$\text{COMP}_n(\mathcal{M}) = \int_{x^n} P(x^n | \hat{\mu}(x^n)) dx^n = \infty,$$

where we abbreviated  $dx_1 \dots dx_n$  to  $dx^n$ . Therefore, we cannot use basic MDL model selection. It also turns out that  $I(\mu) = \sigma^{-2}$  so that

$$\int_{\Theta} \sqrt{|I(\theta)|} d\theta = \int_{\mu \in \mathbb{R}} \sqrt{|I(\mu)|} d\mu = \infty.$$

Thus, the Bayesian universal model approach with Jeffreys' prior cannot be applied either. Does this mean that our MDL model selection and complexity definitions break down even in such a simple case? Luckily, it turns out that they can be repaired, as we now show. In [9] and [33] it is shown that, for all intervals  $[a, b]$ ,

$$\int_{\substack{x^n \\ \hat{\mu}(x^n) \in [a, b]}} P(x^n | \hat{\mu}(x^n)) dx^n = \frac{b-a}{\sqrt{2\pi}\sigma} \cdot \sqrt{n}. \quad (4.33)$$

Suppose for the moment that it is known that  $\hat{\mu}$  lies in some interval  $[-K, K]$  for some fixed  $K$ . Let  $\mathcal{M}_K$  be the set of conditional distributions obtained as follows:  $\mathcal{M}_K = \{P'(\cdot | \mu) | \mu \in \mathbb{R}\}$ , where  $P'(x^n | \mu)$  is the density of  $x^n$  according to the normal distribution with mean  $\mu$ , conditioned on  $|n^{-1} \sum x_i| \leq K$ . By (4.33), the 'conditional' minimax regret distribution  $\bar{P}_{nml}(\cdot | \mathcal{M}_K)$  is well-defined for all  $K > 0$ . That is, for all  $x^n$  with  $|\hat{\mu}(x^n)| \leq K$ , we obtain

$$\bar{P}_{nml}(x^n | \mathcal{M}_K) = \frac{P'(x^n | \hat{\mu}(x^n))}{\int_{|\hat{\mu}(x^n)| < K} P'(x^n | \hat{\mu}(x^n)) dx^n},$$

with regret (or in this case 'conditional' complexity),

$$\text{COMP}_n(\mathcal{M}_K) = \log \int_{|\hat{\mu}(x^n)| < K} P'(x^n | \hat{\mu}(x^n)) dx^n = \log K + \frac{1}{2} \log \frac{n}{2\pi} - \log \sigma + 1.$$

This suggests to redefine the complexity of the full model  $\mathcal{M}$  so that its regret depends on the area in which  $\hat{\mu}$  falls. The most straightforward way of achieving this is to define a meta-universal model for  $\mathcal{M}$ , combining the NML with a two-part code: we encode data by first encoding some value for  $K$ . We then encode the actual data  $x^n$  using the code  $\bar{P}_{nml}(\cdot | \mathcal{M}_K)$ . The resulting code  $\bar{P}_{meta}$  is a universal code for  $\mathcal{M}$  with lengths

$$-\log \bar{P}_{meta}(x^n | \mathcal{M}) := \min_K \{-\log \bar{P}_{meta}(x^n | \mathcal{M}_K) + L(K)\}. \quad (4.34)$$

The idea is now to base MDL model selection on  $\bar{P}_{meta}(\cdot | \mathcal{M})$  as in (4.34) rather than on the (undefined)  $\bar{P}_{nml}(\cdot | \mathcal{M})$ . To make this work, we need to choose  $L$  in a clever manner. A good choice is to encode  $K' = \log K$  as an integer, using the standard code for the integers. To see why, note that the regret of  $\bar{P}_{meta}$  now becomes:

$$\begin{aligned} \mathcal{R} &= -\log \bar{P}_{meta}(x^n | \mathcal{M}) - (-\log P(x^n | \hat{\mu}(x^n))) \\ &= \min_{K, \log K \in \{1, 2, \dots\}} \{\log K + \frac{1}{2} \log \frac{n}{2\pi} - \log \sigma + 1 + 2 \log \lceil \log K \rceil\} + 1 \\ &\leq \log |\hat{\mu}(x^n)| + 2 \log \log |\hat{\mu}(x^n)| + \frac{1}{2} \log \frac{n}{2\pi} - \log \sigma + 4 \\ &\leq \text{COMP}_n(\mathcal{M}_{|\hat{\mu}|}) + 2 \log \text{COMP}_n(\mathcal{M}_{|\hat{\mu}|}) + 3. \end{aligned} \quad (4.35)$$

If we had known a good bound  $K$  on  $|\hat{\mu}|$  *a priori*, we could have used the NML model  $\bar{P}_{nml}(\cdot|\mathcal{M}_K)$ . With ‘maximal’ *a priori* knowledge, we would have used the model  $\bar{P}_{nml}(\cdot|\mathcal{M}_{|\hat{\mu}|})$ , leading to regret  $\text{COMP}_n(\mathcal{M}_{|\hat{\mu}|})$ . The regret achieved by  $\bar{P}_{meta}$  is almost as good as this ‘smallest possible regret-with-hindsight’  $\text{COMP}_n(\mathcal{M}_{|\hat{\mu}|})$ : the difference is much smaller than, in fact logarithmic in,  $\text{COMP}_n(\mathcal{M}_{|\hat{\mu}|})$  itself, no matter what  $x^n$  we observe. This is the underlying reason why we choose to encode  $K$  with log-precision: the basic idea in refined MDL was to minimize the worst-case regret, or additional code-length compared to the code that achieves the minimal code-length with hindsight. Here, we use this basic idea on a meta-level: we design a code such that the additional regret is minimized, compared to the code that achieves the minimal regret with hindsight.  $\diamond$

This meta-two-part coding idea was introduced by Rissanen [95]. It can be extended to a wide range of models with  $\text{COMP}_n(\mathcal{M}) = \infty$ ; for example, if the  $X_i$  represent outcomes of a Poisson or geometric distribution, one can encode a bound on  $\mu$  just like in example 4.18. If  $\mathcal{M}$  is the full Gaussian model with both  $\mu$  and  $\sigma^2$  allowed to vary, one has to encode a bound on  $\hat{\mu}$  and a bound on  $\hat{\sigma}^2$ . Essentially the same holds for linear regression problems, section 4.7.

**Renormalized maximum likelihood** Meta two-part coding is just one possible solution to the problem of undefined  $\text{COMP}_n(\mathcal{M})$ . It is suboptimal, the main reason being the use of 2-part codes. Indeed, these 2-part codes are not complete (section 4.1): they reserve several code words for the same data  $D = (x_1, \dots, x_n)$  (one for each integer value of  $\log K$ ). Therefore, there must exist more efficient (one-part) codes  $\bar{P}'_{meta}$  such that for all  $x^n \in \mathcal{X}^n$ ,  $\bar{P}'_{meta}(x^n) > \bar{P}_{meta}(x^n)$ . In accordance with the idea that we should minimize description length, such alternative codes are preferable. This realization has led to a search for more efficient and intrinsic solutions to the problem. In [33], the possibility is considered of restricting the parameter values rather than the data, and develop a general framework for comparing universal codes for models with undefined  $\text{COMP}(\mathcal{M})$ . Rissanen [97] suggests the following elegant solution. He defines the renormalized maximum likelihood (RNML) distribution  $\bar{P}_{rnml}$ . In our Gaussian example, this universal model would be defined as follows. Let  $\hat{K}(x^n)$  be the bound on  $\hat{\mu}(x^n)$  that maximizes  $\bar{P}_{nml}(x^n|\mathcal{M}_K)$  for the actually given  $K$ . That is,  $\hat{K}(x^n) = |\hat{\mu}(x^n)|$ . Then  $\bar{P}_{rnml}$  is for all  $x^n \in \mathcal{X}^n$  defined as,

$$\bar{P}_{rnml}(x^n|\mathcal{M}) = \frac{\bar{P}_{nml}(x^n|\mathcal{M}_{\hat{K}(x^n)})}{\int_{x^n \in \mathbb{R}^n} \bar{P}_{nml}(x^n|\mathcal{M}_{\hat{K}(x^n)}) dx^n}. \quad (4.36)$$

Model selection between a finite set of models now proceeds by selecting the model maximizing the re-normalized likelihood (4.36).

**Region indifference** All the approaches considered thus far slightly prefer some regions of the parameter space over others. In spite of its elegance, even the Rissanen renormalization is slightly ‘arbitrary’ in this way: had we chosen the origin of the real axis differently, the same sequence  $x^n$  would have achieved a different code length  $-\log \bar{P}_{rnml}(x^n|\mathcal{M})$ . In recent work, Liang and Barron [73], [74] consider a novel and quite different approach for dealing with infinite  $\text{COMP}_n(\mathcal{M})$  that partially addresses this problem. They make use of the fact that, while Jeffreys’ prior is improper ( $\int \sqrt{|I(\theta)|} d\theta$  is infinite), using Bayes’ rule we can still compute Jeffreys’ posterior

based on the first few observations, and this posterior turns out to be a proper probability measure after all. Liang and Barron use universal models of a somewhat different type than  $\bar{P}_{nml}$ , so it remains to be investigated whether their approach can be adapted to the form of MDL discussed here.

### 4.6.3 The general picture

Section 4.6.1 illustrates that, in all applications of MDL, we first define a single universal model that allows us to code all sequences with length equal to the given sample size. If the set of models is finite, we use the uniform prior. We do this in order to be as ‘honest’ as possible, treating all models under consideration on the same footing. But if the set of models becomes infinite, there exists no uniform prior any more. Therefore, we must choose a non-uniform prior/non-fixed length code to encode the model index. In order to treat all models still ‘as equally as possible’, we should use some code which is ‘close’ to uniform, in the sense that the code length increases only very slowly with  $k$ . We choose the standard prior for the integers (example 4.4), but we could also have chosen different priors, for example, a prior  $P(k)$  which is uniform on  $k = 1 \dots M$  for some large  $M$ , and  $P(k) \propto k^{-2}$  for  $k > M$ . Whatever prior we choose, we are forced to encode a slight preference of some models over others; see section 4.9.1.

#### Observation 4.4 (General ‘refined’ MDL principle for model selection)

Suppose we plan to select between models  $\mathcal{M}^{(1)}, \mathcal{M}^{(2)}, \dots$  for data  $D = (x_1, \dots, x_n)$ . MDL tells us to design a universal code  $\bar{P}$  for  $\mathcal{X}^n$ , in which the index  $k$  of  $\mathcal{M}^{(k)}$  is encoded explicitly. The resulting code has two parts, the two sub-codes being defined such that:

1. All models  $\mathcal{M}^{(k)}$  are treated on the same footing, as far as possible: we assign a uniform prior to these models, or, if that is not a possible, a prior ‘close to’ uniform.
2. All distributions within each  $\mathcal{M}^{(k)}$  are treated on the same footing, as far as possible: we use the minimax regret universal model  $\bar{P}_{nml}(x^n | \mathcal{M}^{(k)})$ . If this model is undefined or too hard to compute, we instead use a different universal model that achieves regret ‘close to’ the minimax regret for each submodel of  $\mathcal{M}^{(k)}$  in the sense of (4.35).

In the end, we encode data  $D$  using a hybrid two-part/one-part universal model, explicitly encoding the models we want to select between and implicitly encoding any distributions contained in those models.

Section 4.6.2 applies the same idea, but implemented at a meta-level: we try to associate with  $\mathcal{M}^{(k)}$  a code for encoding outcomes in  $\mathcal{X}^n$  that achieves uniform (= minimax) regret for every sequence  $x^n$ . If this is not possible, we still try to assign regret as ‘uniformly’ as we can, by carving up the parameter space in regions with larger and larger minimax regret, and devising a universal code that achieves regret not much larger than the minimax regret achievable within the smallest region containing the ML estimator. Again, the codes we used, encoded a slight preference of some regions of the parameter space over others, but our aim was to keep this preference as small as possible. The general idea is summarized in observation 4.4, which provides an (informal) definition of MDL, but only in a restricted context. If we go beyond that context,

these prescriptions cannot be used literally - but extensions in the same spirit suggest themselves. Here is a first example of such an extension:

**Example 4.19 (MDL and Local Maxima in the Likelihood)**

In practice we often work with models for which the ML estimator cannot be calculated efficiently; or at least, no algorithm for efficient calculation of the ML estimator is known. Examples are finite and Gaussian mixtures and hidden Markov models. In such cases one typically resorts to methods such as expectation maximization (EM) or gradient descent, which find a local maximum of the likelihood surface (function)  $P(x^n|\theta)$ , leading to a local maximum likelihood estimator (LML)  $\hat{\theta}(x^n)$ . Suppose we need to select between a finite number of such models. We may be tempted to pick the model  $\mathcal{M}$  maximizing the normalized likelihood  $\bar{P}_{nml}(x^n|\mathcal{M})$ . However, if we then plan to use the local estimator  $\hat{\theta}(x^n)$  for predicting future data, this is not the right thing to do. To see this, note that, if suboptimal estimators  $\hat{\theta}$  are to be used, the ability of model  $\mathcal{M}$  to fit arbitrary data patterns may be severely diminished! Rather than using  $\bar{P}_{nml}$ , we should redefine it to take into account the fact that  $\hat{\theta}$  is not the global ML estimator:

$$\bar{P}'_{nml}(x^n) := \frac{P(x^n|\hat{\theta}(x^n))}{\sum_{x^n \in \mathcal{X}^n} P(x^n|\hat{\theta}(x^n))},$$

leading to an adjusted parametric complexity

$$\text{COMP}'_n(\mathcal{M}) := \log \sum_{x^n \in \mathcal{X}^n} P(x^n|\hat{\theta}(x^n)), \quad (4.37)$$

which, for every estimator  $\hat{\theta}$  different from  $\hat{\theta}$  must be strictly smaller than  $\text{COMP}_n(\mathcal{M})$ .  
 $\diamond$

**Summary** We have shown how to extend refined MDL beyond the restricted settings of section 4.5. This uncovered the general principle behind refined MDL for model selection, given in observation 4.4. General as it may be, it only applies to model selection - in the next section we briefly discuss extensions to other applications.

## 4.7 Beyond parametric model selection

The general principle as given in observation 4.4 only applies to model selection. It can be extended in several directions. These range over many different tasks of inductive inference - we mention prediction, transduction (as defined in [115]), clustering [68] and similarity detection [72]. In these areas there has been less research and a 'definite' MDL approach has not yet been formulated.

MDL has been developed in some detail for some other inductive tasks: non-parametric inference, parameter estimation and regression and classification problems. We give a very brief overview of these - for details we refer to [9], [52] and, for the classification case [49].

**Non-parametric inference** Sometimes the model class  $\mathcal{M}$  is so large that it cannot be finitely parameterized. For example, let  $\mathcal{X} = [0, 1]$  be the unit interval and let  $\mathcal{M}$  be the i.i.d. model consisting of all distributions on  $\mathcal{X}$  with densities  $f$  such that  $-\log f(x)$  is



a continuous function on  $\mathcal{X}$ .  $\mathcal{M}$  is clearly ‘non-parametric’: it cannot be meaningfully parameterized by a connected finite-dimensional parameter set  $\Theta^{(k)} \subseteq \mathbb{R}^k$ . We may still try to learn a distribution from  $\mathcal{M}$  in various ways, for example by histogram density estimation [94] or kernel density estimation [93]. MDL is quite suitable for such applications, in which we typically select a density  $f$  from a class  $\mathcal{M}^{(n)} \subset \mathcal{M}$ , where  $\mathcal{M}^{(n)}$  grows with  $n$ , and every  $P^* \in \mathcal{M}$  can be arbitrarily well approximated by members of  $\mathcal{M}^{(n)}, \mathcal{M}^{(n+1)}, \dots$  in the sense that [9]

$$\lim_{n \rightarrow \infty} \inf_{P \in \mathcal{M}^{(n)}} D(P^* || P) = 0.$$

Here  $D$  is the Kullback-Leibler divergence [20] between  $P^*$  and  $P$ .

**MDL parameter estimation: three approaches** The ‘crude’ MDL method (section 4.3) was a means of doing model selection and parameter estimation at the same time. ‘Refined’ MDL only dealt with selection of models. If instead, or at the same time, parameter estimates are needed, they may be obtained in three different ways. Historically the first way [93], [52] was to simply use the refined MDL principle to pick a parametric model  $\mathcal{M}^{(k)}$ , and then, within  $\mathcal{M}^{(k)}$ , pick the ML estimator  $\hat{\theta}^{(k)}$ . After all, we associate with  $\mathcal{M}^{(k)}$  the distribution  $\bar{P}_{nml}$  with code lengths ‘as close as possible’ to those achieved by the ML estimator. This suggests that within  $\mathcal{M}^{(k)}$ , we should prefer the ML estimator. But upon closer inspection, observation 4.4 suggests to use a two-part code also to select  $\theta$  within  $\mathcal{M}^{(k)}$ ; namely, we should discretize the parameter space in such a way that the resulting 2-part code achieves the minimax regret among all two-part codes; we then pick the (quantized)  $\theta$  minimizing the two-part code length. Essentially this approach has been worked out in detail by Barron and Cover [8]. The resulting estimators may be called two-part code MDL estimators. A third possibility is to define predictive MDL estimators such as the Laplace and Jeffreys estimators of example 4.17; once again, these can be understood as an extension of observation 4.4 [9]. These second and third possibilities are more sophisticated than the first. However, if the model  $\mathcal{M}$  is finite-dimensional parametric and  $n$  is large, then both the two-part and the predictive MDL estimators will become indistinguishable from the maximum likelihood estimators. For this reason, it has sometimes been claimed that MDL parameter estimation is just ML parameter estimation. Since for small samples, the estimates can be quite different, this statement is misleading.

**Regression** In regression problems we are interested in learning how the values  $y_1, \dots, y_n$  of a regression variable  $Y$  depend on the values  $x_1, \dots, x_n$  of the regressor variable  $X$ . We assume or hope that there exists some function  $h : \mathcal{X} \rightarrow \mathcal{Y}$  so that  $h(X)$  predicts the value  $Y$  reasonably well, and we want to learn such an  $h$  from data. To this end, we assume a set of candidate predictors (functions)  $\mathcal{H}$ . In example 3.2, we took  $\mathcal{H}$  to be the set of all polynomials. In the standard formulation of this problem, we take  $h$  to express that

$$Y_i = h(X_i) + Z_i, \quad (4.38)$$

where the  $Z_i$  are i.i.d. Gaussian random variables with mean 0 and some variance  $\sigma^2$ , independent of  $X_i$ . That is, we assume Gaussian noise: equation (4.38) implies that the conditional density of  $y_1, \dots, y_n$ , given  $x_1, \dots, x_n$ , is equal to the product of  $n$  Gaussian densities

$$P(y^n | x^n, \sigma, h) = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp \left( - \frac{\sum_{i=1}^n (y_i - h(x_i))^2}{2\sigma^2} \right). \quad (4.39)$$

With this choice, the log-likelihood becomes a linear function of the squared error:

$$-\ln P(y^n|x^n, \sigma, h) = \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - h(x_i))^2 + \frac{n}{2} \ln 2\pi\sigma^2. \quad (4.40)$$

Note that  $\ln(\cdot) = \ln 2 \log(\cdot)$ . Let us now assume that  $\mathcal{H} = \bigcup_{k \geq 1} \mathcal{H}^{(k)}$  where for each  $k$ ,  $\mathcal{H}^{(k)}$  is a set of functions  $h : \mathcal{X} \rightarrow \mathcal{Y}$ . For example,  $\mathcal{H}^{(k)}$  may be the set of  $k$ -th degree polynomials.

With each model  $\mathcal{H}^{(k)}$  we can associate a set of densities (4.39), one for each  $(h, \sigma^2)$  with  $h \in \mathcal{H}^{(k)}$  and  $\sigma^2 \in \mathbb{R}^+$ . Let  $\mathcal{M}^{(k)}$  be the resulting set of conditional distributions. Each  $P(\cdot|h, \sigma^2) \in \mathcal{M}^{(k)}$  is identified by the parameter vector  $(\alpha_0, \dots, \alpha_k, \sigma^2)$  so that  $h(x) := \sum_{j=0}^k \alpha_j x^j$ . By section 4.6.1, equation (4.8), MDL tells us to select the model minimizing

$$-\ln \bar{P}(y^n|\mathcal{M}^{(k)}, x^n) + L(k), \quad (4.41)$$

where we may take  $L(k) = 2 \log k + 1$ , and  $\bar{P}(\cdot|\mathcal{M}^{(k)}, \cdot)$  is now a conditional universal model with small minimax regret. Equation (4.41) ignores the code length of  $x_1, \dots, x_n$ . Intuitively, this is because we are only interested in learning how  $y$  depends on  $x$ ; therefore, we do not care how many bits are needed to encode  $x$ . Formally, this may be understood as follows: we really are encoding the  $x$ -values as well, but we do so using a fixed code that does not depend on the hypothesis  $h$  under consideration. Thus, we are really trying to find the model  $\mathcal{M}^{(k)}$  minimizing

$$-\ln \bar{P}(y^n|\mathcal{M}^{(k)}, x^n) + L(k) + L'(x^n),$$

where  $L'$  represents some code for  $\mathcal{X}^n$ . Since this code length does not involve  $k$ , it can be dropped from the minimization; see observation 4.5. We will not go into the precise definition of  $\bar{P}(y^n|\mathcal{M}^{(k)}, x^n)$ . Ideally, it should be an NML distribution, but just as in example 4.18, this NML distribution is not well-defined. We can get reasonable alternative universal models after all, using any of the methods described in section 4.6.2; see [9] and [96] for details.

**Observation 4.5 (When the code length for  $x^n$  can be ignored)**

*If all models under consideration represent conditional densities or probability mass functions  $P(Y|X)$ , then the code length for  $X_1, \dots, X_n$  can be ignored in model and parameter selection. Examples are applications of MDL in classification and regression.*

**‘Non-probabilistic’ regression and classification** In the approach we just described, we modeled the noise as being normally distributed. Alternatively, it has been tried to directly try to learn functions  $h \in \mathcal{H}$  from the data, without making any probabilistic assumptions about the noise [93], [7], [125], [44] and [45]. The idea is to learn a function  $h$  that leads to good predictions of future data from the same source in the spirit of Vapnik’s [115] statistical learning theory. Here prediction quality is measured by some fixed loss function; different loss functions lead to different instantiations of the procedure. Such a version of MDL is meant to be more robust, leading to inference of a ‘good’  $h \in \mathcal{H}$  irrespective of the details of the noise distribution. This loss-based

approach has also been the method of choice in applying MDL to classification problems. Here  $Y$  takes on values in a finite set, and the goal is to match each feature  $X$  (for example, a bit map of a handwritten digit) with its corresponding label or class (e.g., a digit). While several versions of MDL for classification have been proposed [83], [93], [64], most of these can be reduced to the same approach based on a 0/1-valued loss function [44]. In recent work [49] it is shown that this MDL approach to classification without making assumptions about the noise may behave suboptimally: situations where no matter how large  $n$ , MDL keeps overfitting, selecting an overly complex model with suboptimal predictive behavior. Modifications of MDL suggested by Barron [7] and Yamanishi [125] do not suffer from this defect, but they do not admit a natural coding interpretation any longer. All in all, current versions of MDL that avoid probabilistic assumptions are still in their infancy, and more research is needed to find out whether they can be modified to perform well in more general and realistic settings.

**Summary** In the previous sections, we have covered basic refined MDL (section 4.5), general refined MDL (section 4.6), and several extensions of refined MDL (this section). This concludes our technical description of refined MDL. It only remains to place MDL in its proper context: what does it do compared to other methods of inductive inference? And how well does it perform, compared to other methods? The next two sections are devoted to these questions.

## 4.8 Relations to other approaches to inductive inference

How does MDL compare to other model selection and statistical inference methods? In order to answer this question, we first have to be precise about what we mean by 'MDL'; this is done in section 4.8.1. We then continue in section 4.8.2 by summarizing MDL's relation to Bayesian inference, Wallace's MML Principle, Dawid's prequential model validation, cross-validation and an 'idealized' version of MDL based on Kolmogorov complexity. The literature has also established connections between MDL and Jaynes' [57] maximum entropy principle [29], [71], [44], [46], [49] and Vapnik's [115] structural risk minimization principle [44]. Relations between MDL and Akaike's AIC [14] are subtle. They are discussed in [108].

### 4.8.1 What is MDL?

'MDL' is used by different authors in somewhat different meanings. Some authors use MDL as a broad umbrella term for all types of inductive inference based on data compression. This would, for example, include the 'idealized' versions of MDL based on Kolmogorov complexity and Wallaces's MML Principle, to be discussed below. On the other extreme, for historical reasons, some authors use the MDL criterion to describe a very specific (and often not very successful) model selection criterion equivalent to BIC, discussed further below.

Here we adopt the meaning of the term that is embraced in the survey [9], written by arguably the three most important contributors to the field: MDL for general inference based on universal models. These include, but are not limited to approaches in the spirit of observation 4.4. For example, some authors have based their inferences

on ‘expected’ rather than ‘individual sequence’ universal models [9], [73]. Moreover, if we go beyond model selection (section 4.7), then the ideas of observation 4.4 have to be modified to some extent. In fact, one of the main strengths of ‘MDL’ in this broad sense is that it can be applied to ever more exotic modeling situations, in which the models do not resemble anything that is usually encountered in statistical practice. An example is the model of context-free grammars, already suggested by Solomonoff [106]. In this chapter, we call applications of MDL that strictly fit into the scheme of observation 4.4 refined MDL for model/hypothesis selection; when we simply say ‘MDL’, we mean ‘inductive inference based on universal models’. This form of inductive inference goes hand in hand with Rissanen’s radical MDL philosophy, which views learning as finding useful properties of the data, not necessarily related to the existence of a ‘truth’ underlying the data. This view was outlined in chapter 3, section 3.5. Although MDL practitioners and theorists are usually sympathetic to it, the different interpretations of MDL listed in section 4.5 make clear that MDL applications can also be justified without adopting such a radical philosophy.

#### 4.8.2 MDL and Bayesian inference

Bayesian statistics [70], [10] is one of the most well-known, frequently and successfully applied paradigms of statistical inference. It is often claimed that ‘MDL is really just a special case of Bayes\*’. Although there are close similarities, this is simply not true. To see this quickly, consider the basic quantity in refined MDL: the NML distribution  $\bar{P}_{nml}$ , equation (4.18). While  $\bar{P}_{nml}$  - although defined in a completely different manner - turns out to be closely related to the Bayesian marginal likelihood, this is no longer the case for its ‘localized’ version (4.37). There is no mention of anything like this code/distribution in any Bayesian textbook! Consequently, it must be the case that Bayes and MDL are somehow different.

**MDL as a maximum probability principle** For a more detailed analysis, we need to distinguish between the two central tenets of modern Bayesian statistics: (1) Probability distributions are used to represent uncertainty, and to serve as a basis for making predictions; rather than standing for some imagined ‘true state of nature’. (2) All inference and decision-making is done in terms of prior and posterior distributions. MDL sticks with (1) (although here the ‘distributions’ are primarily interpreted as ‘code length functions’), but not (2): MDL allows the use of arbitrary universal models such as NML and prequential universal models; the Bayesian universal model does not have a special status among these. In this sense, Bayes offers the statistician less freedom in choice of implementation than MDL. In fact, MDL may be reinterpreted as a maximum probability principle, where the maximum is relative to some given model, in the worst-case over all sequences (Rissanen [92], [93] uses the phrase ‘global maximum likelihood principle’). Thus, whenever the Bayesian universal model is used in an MDL application, a prior should be used that minimizes worst-case code length regret, or equivalently, maximizes worst-case relative probability. There is no comparable principle for choosing priors in Bayesian statistics, and in this respect, Bayes offers a lot more freedom than MDL.

\* The reasons are probably historical: while the underlying philosophy has always been different, most actual implementations of MDL ‘looked’ quite Bayesian until Rissanen introduced the use of  $\bar{P}_{nml}$ .

**Example 4.20**

There is a conceptual problem with Bayes' use of prior distributions: in practice, we very often want to use models which we *a priori* know to be wrong - see example 3.5. If we use Bayes for such models, then we are forced to put a prior distribution on a set of distributions which we know to be wrong. From an MDL viewpoint, these priors are interpreted as tools to achieve short code lengths rather than degrees-of-belief and there is nothing strange about the situation; but from a Bayesian viewpoint, it seems awkward. To be sure, Bayesian inference often gives good results even if the model  $\mathcal{M}$  is known to be wrong; the point is that (a) if one is a strict Bayesian, one would never apply Bayesian inference to such misspecified  $\mathcal{M}$ , and (b), the Bayesian theory offers no clear explanation of why Bayesian inference might still give good results for such  $\mathcal{M}$ . MDL provides both code length and predictive sequential interpretations of Bayesian inference, which help explain why Bayesian inference may do something reasonable even if  $\mathcal{M}$  is misspecified. To be fair, we should add that there exists variations of the Bayesian philosophy (e.g. [26]) which avoid the conceptual problem we just described.  $\diamond$

**MDL and BIC** In the first paper on MDL, Rissanen [88] used a two-part code and showed that, asymptotically, and under regularity conditions, the two-part code length of  $x^n$  based on a  $k$ -parameter model  $\mathcal{M}$  with an optimally discretized parameter space is given by

$$L = -\log P(x^n | \hat{\theta}(x^n)) + \frac{k}{2} \log n. \quad (4.42)$$

Note that the  $\mathcal{O}(1)$ -terms are ignored. However, as we have already seen, they can be quite important. In the same year Schwarz [100] showed that, for large enough  $n$ , Bayesian model selection between two exponential families amounts to selecting the model minimizing (4.42), ignoring  $\mathcal{O}(1)$ -terms as well. As a result of Schwarz's paper, model selection based on (4.42) became known as the BIC (Bayesian information criterion). Not taking into account the functional form of the model  $\mathcal{M}$ , it often does not work very well in practice.

It has sometimes been claimed that MDL = BIC; for example, in [14], page 286 it is written 'Rissanen's result is equivalent to BIC'. This is wrong, even for the 1989 version of MDL that is referred to - as pointed out in [34], the BIC approximation only holds if the number of parameters  $k$  is kept fixed and  $n$  goes to infinity. If we select between nested families of models where the maximum number of parameters  $k$  considered is either infinite or grows with  $n$ , then model selection based on both  $\bar{P}_{mdl}$  and on  $\bar{P}_{Bayes}$  tends to select quite different models than BIC - if  $k$  gets closer to  $n$ , the contribution to  $\text{COMP}_n(\mathcal{M})$  of each additional parameter becomes much smaller than  $0.5 \log n$  [34]. However, researchers who claim MDL = BIC have a good excuse: in early work, Rissanen himself has used the phrase 'MDL criterion' to refer to (4.42), and unfortunately, the phrase has stuck.

**MDL and MML** MDL shares some ideas with the minimum message length (MML) principle which predates MDL by 10 years. Key references are [121], [122] and [123], a long list is presented in [19]. Just as in MDL, MML chooses the hypothesis minimizing the code-length of the data. But the codes that are used are quite different from those in MDL. First of all, in MML one always uses two-part codes, so that MML automatically selects both a model family and parameter values. Second, while the

MDL codes such as  $\bar{P}_{nml}$  minimize worst-case relative code-length (regret), the two-part codes used by MML are designed to minimize expected absolute code-length. Here the expectation is taken over a subjective prior distribution defined on the collection of models and parameters under consideration. While this approach contradicts Rissanen's philosophy, in practice it often leads to similar results.

Indeed, Wallace and his co-workers, [121], [122] and [123], stress that their approach is fully (subjective) Bayesian. Strictly speaking, a Bayesian should report his findings by citing the full posterior distribution. But sometimes one is interested in a single model, or hypothesis for the data. A good example is the inference of phylogenetic trees in biological applications: the full posterior would consist of a mixture of several of such trees, which might all be quite different from each other. Such a mixture is almost impossible to interpret - to get insight in the data we need a single tree. In that case, Bayesians often use the MAP (maximum a posteriori) hypothesis which maximizes the posterior, or the posterior mean parameter value. The first approach has some unpleasant properties, for example, it is not invariant under reparameterization. The posterior mean approach cannot be used if different model families are to be compared with each other. The MML method provides a theoretically sound way of proceeding in such cases.

#### 4.8.3 MDL, prequential analysis and cross validation

In a series of papers, [22], [23], [24] Dawid put forward a methodology for probability and statistics based on sequential prediction which he called the prequential approach. When applied to model selection problems, it is closely related to MDL - Dawid proposes to construct, for each model  $\mathcal{M}^{(j)}$  under consideration, a 'probability forecasting system' (a sequential prediction strategy) where the  $i + 1$ -st outcome is predicted based on either the Bayesian posterior  $\bar{P}_{Bayes}(\theta|x^i)$  or on some estimator  $\hat{\theta}(x^i)$ . Then the model is selected for which the associated sequential prediction strategy minimizes the accumulated prediction error. Related ideas were put forward in [55] under the name forward validation and in [90]. From section 4.5.4 we see that these are just forms of MDL - strictly speaking, every universal code can be thought of as as prediction strategy, but for the Bayesian and the plug-in universal models (sections 4.5.3 and 4.5.4) the interpretation is much more natural than for others<sup>†</sup>. Dawid mostly talks about such 'predictive' universal models. On the other hand, Dawid's framework allows to adjust the prediction loss to be measured in terms of arbitrary loss functions, not just the log loss. In this sense, it is more general than MDL. Finally, the prequential idea goes beyond statistics: there is also a 'prequential approach' to probability theory developed by Dawid [25] and Shafer and Vovk [101].

Note that the prequential approach is similar in spirit to cross-validation. In this sense MDL is related to cross-validation as well. The main differences are that in MDL and the prequential approach, (1) all predictions are done sequentially (the future is never used to predict the past), and (2) each outcome is predicted exactly once.

<sup>†</sup> The reason is that Bayesian and plug-in models can be interpreted as probabilistic sources. The NML and the two-part code models are no probabilistic sources, since  $\bar{P}^{(n)}$  and  $\bar{P}^{(n+1)}$  are not compatible in the sense of section 4.1.

#### 4.8.4 Kolmogorov complexity and structure function: ideal MDL

Kolmogorov complexity [71] has played a large but mostly inspirational role in Rissanen's development of MDL. Over the last fifteen years, several 'idealized' versions of MDL have been proposed, which are more directly based on Kolmogorov complexity theory [6], [8], [71], [116]. These are all based on two-part codes, where hypotheses are described using a universal programming language such as C or Pascal. For example, in one proposal [8], given data  $D$  one picks the distribution minimizing

$$K(P) + (-\log P(D)), \quad (4.43)$$

where the minimum is taken over all computable probability distributions, and  $K(P)$  is the length of the shortest computer program that, when input  $(x, d)$ , outputs  $P(x)$  to  $d$  bits precision. While such a procedure is mathematically well-defined, it cannot be used in practice. The reason is that in general, the  $P$  minimizing (4.43) cannot be effectively computed. Kolmogorov himself used a variation of (4.43) in which one adopts, among all  $P$  with  $K(P) - \log P(D) \approx K(D)$ , the  $P$  with smallest  $K(P)$ . Here  $K(D)$  is the Kolmogorov complexity of  $D$ , that is, the length of the shortest computer program that prints  $D$  and then halts. This approach is known as the Kolmogorov structure function or minimum sufficient statistic approach [120]. In this approach, the idea of separating data and noise (section 4.5.1) is taken as basic, and the hypothesis selection procedure is defined in terms of it. The selected hypothesis may now be viewed as capturing all structure inherent in the data - given the hypothesis, the data cannot be distinguished from random noise. Therefore, it may be taken as a basis for lossy data compression - rather than sending the whole sequence, one only sends the hypothesis representing the 'structure' in the data. The receiver can then use this hypothesis to generate 'typical' data for it - this data should then 'look just the same' as the original data  $D$ . Rissanen views this separation idea as perhaps the most fundamental aspect of 'learning by compression'. Therefore, in recent work he has tried to relate MDL (as defined here, based on lossless compression) to the Kolmogorov structure function (thereby connecting it to lossy compression) and, as he puts it, 'opening up a new chapter in the MDL theory' [116], [120], [99].

**Summary** We have shown that MDL is closely related, yet distinct from, to several other methods for inductive inference. In the next section we discuss how well it performs compared to such other methods.

### 4.9 Problems for MDL?

Some authors have criticized MDL either on conceptual grounds (the idea makes no sense) [124], [27] or on practical grounds (sometimes it does not work very well in practice) [64], [82]. Are these criticisms justified? Let us consider them in turn.

#### 4.9.1 Conceptual problems: Occam's razor

The most-often heard conceptual criticisms are invariably related to Occam's razor. We have already discussed in section 3.5 of the previous chapter why we regard these criticisms as being entirely mistaken. Based on our newly acquired technical knowledge of MDL, let us discuss these criticisms a little bit further:

**1. ‘Occam’s razor (and MDL) is arbitrary’** If we restrict ourselves to refined MDL for comparing a finite number of models for which the NML distribution is well-defined, then there is nothing arbitrary about MDL - it is exactly clear what codes we should use for our inferences. The NML distribution and its close cousins, the Jeffreys’ prior marginal likelihood  $\bar{P}_{Bayes}$  and the asymptotic expansion (4.21) are all invariant to continuous 1-to-1 reparameterizations of the model: parameterizing our model in a different way (choosing a different ‘description language’) does not change the inferred description lengths.

If we go beyond models for which the NML distribution is defined, and/or we compare an infinite set of models at the same time, then some ‘subjectivity’ is introduced - while there are still tough restrictions on the codes that we are allowed to use, all such codes prefer some hypotheses in the model over others. If one does not have an *a priori* preference over any of the hypotheses, one may interpret this as some arbitrariness being added to the procedure. But this ‘arbitrariness’ is of an infinitely milder sort than the arbitrariness that can be introduced if we allow completely arbitrary codes for the encoding of hypotheses as in crude two-part code MDL, section 4.3.

Things get more subtle if we are interested not only in model selection (find the best order Markov chain for the data) but also in infinite-dimensional estimation (find the best Markov chain parameters for the data, among the set  $\mathcal{B}$  of all Markov chains of each order). In the latter case, if we are to apply MDL, we somehow have to carve up  $\mathcal{B}$  into subsets  $\mathcal{M}^{(0)} \subseteq \mathcal{M}^{(1)} \subseteq \dots \subseteq \mathcal{B}$ . Suppose that we have already chosen  $\mathcal{M}^{(1)} = \mathcal{B}^{(1)}$  as the set of 1-st order Markov chains. We normally take  $\mathcal{M}^{(0)} = \mathcal{B}^{(0)}$ , the set of 0-th order Markov chains (Bernoulli distributions). But we could also have defined  $\mathcal{M}^{(0)}$  as the set of all 1-st order Markov chains with  $P(X_{i+1} = 1 | X_i = 1) = P(X_{i+1} = 0 | X_i = 0)$ . This defines a one-dimensional subset of  $\mathcal{B}^{(1)}$  that is not equal to  $\mathcal{B}^{(0)}$ . While there are several good reasons\* for choosing  $\mathcal{B}^{(0)}$  rather than  $\mathcal{M}^{(0)}$ , there may be no indication that  $\mathcal{B}^{(0)}$  is somehow *a priori* more likely than  $\mathcal{M}^{(0)}$ . While MDL tells us that we somehow have to carve up the full set  $\mathcal{B}$ , it does not give us precise guidelines on how to do this - different carvings may be equally justified and lead to different inferences for small samples. In this sense, there is indeed some form of arbitrariness in this type of MDL applications. But this is unavoidable: we stress that this type of arbitrariness is enforced by all combined model/parameter selection methods - whether they be of the structural risk minimization type [115], AIC-type [14], cross-validation or any other type. The only alternative is treating all hypotheses in the huge class  $\mathcal{B}$  on the same footing, which amounts to maximum likelihood estimation and extreme overfitting.

**2. ‘Occam’s razor is false’** We often try to model real-world situations that can be arbitrarily complex, so why should we favor simple models? We gave an informal answer in chapter 3 where we claimed that even if the true data generating machinery is very complex, it may be a good strategy to prefer simple models for small sample sizes.

\* For example,  $\mathcal{B}^{(0)}$  is better interpretable.



We are now in a position to give one formalization of this informal claim: it is simply the fact that MDL procedures, with their built-in preference for 'simple' models with small parametric complexity, are typically statistically consistent achieving good rates of convergence, whereas methods such as maximum likelihood - which do not take model complexity into account - are typically inconsistent whenever they are applied to complex enough models such as the set of polynomials of each degree or the set of Markov chains of all orders. This has implications for the quality of predictions based on complex enough models, no matter how many training data we observe. If we use the maximum likelihood distribution to predict future data from the same source, the prediction error we make will not converge to the prediction error that could be obtained if the true distribution were known; if we use an MDL submodel/parameter estimate (section 4.7), the prediction error will converge to this optimal achievable error.

Of course, consistency is not the only desirable property of a learning method, and it may be that in some particular settings, and under some particular performance measures, some alternatives to MDL outperform MDL. Indeed this can happen - see below. Yet it remains the case that all methods we know of that successfully deal with models of arbitrary complexity have a built-in preference for selecting simpler models at small sample sizes - methods such as Vapnik's [115] structural risk minimization, penalized minimum error estimators [7] and the Akaike criterion [14] all trade-off complexity with error on the data, the result invariably being that in this way, good convergence properties can be obtained. While these approaches measure 'complexity' in a manner different from MDL, and attach different relative weights to error on the data and complexity, the fundamental idea of finding a trade-off between 'error' and 'complexity' remains.

#### 4.9.2 Practical problems with MDL

We just described some perceived problems about MDL. Unfortunately, there are also some real ones: MDL is not a perfect method. While in many cases, the methods described here perform very well<sup>†</sup> there are also cases where they perform suboptimally compared to other state-of-the-art methods. Often this is due to one of two reasons:

1. An asymptotic formula like (4.21) was used and the sample size was not large enough to justify this [81].
2.  $\bar{P}_{nml}$  was undefined for the models under consideration, and this was solved by cutting off the parameter ranges at ad hoc values [69].

In these cases the problem probably lies in the use of invalid approximations rather than with the MDL idea itself. More research is needed to find out when the asymptotics and other approximations can be trusted, and what is the 'best' way to deal with undefined  $\bar{P}_{nml}$ . For the time being, we suggest to avoid using (4.21) whenever possible, and to never cut off the parameter ranges at arbitrary values - instead, if

<sup>†</sup> We mention [51], [52] reporting excellent behavior of MDL in regression contexts; and [2], [67], [79] reporting excellent behavior of predictive (prequential) coding in Bayesian network model selection and regression. Also, 'objective Bayesian' model selection methods are frequently and successfully used in practice [63]. Since these are based on non-informative priors such as Jeffreys', they often coincide with a version of refined MDL and thus indicate successful performance of MDL.

$\text{COMP}_n(\mathcal{M})$  becomes infinite, then some of the methods described in section 4.6.2 should be used. Given these restrictions,  $\bar{P}_{nml}$  and Bayesian inference with Jeffreys' prior are the preferred methods, since they both achieve the minimax regret. If they are either ill-defined or computationally prohibitive for the models under consideration, one can use a prequential method or a sophisticated two-part code such as described by Barron and Cover [8].

**MDL and misspecification** However, there is a class of problems where MDL is problematic in a more fundamental sense. Namely, if none of the distributions under consideration represents the data generating machinery very well, then both MDL and Bayesian inference may sometimes do a bad job in finding the 'best' approximation within this class of not-so-good hypotheses. This has been observed in practice<sup>†</sup> [64], [17], [82]. In [49] it is shown that MDL can behave quite unreasonably for some classification problems in which the true distribution is not in  $\mathcal{M}$ . This is closely related to the problematic behavior of MDL for classification tasks as mentioned in section 4.7. All this is a bit ironic, since MDL was explicitly designed not to depend on the untenable assumption that some  $P^* \in \mathcal{M}$  generates the data. But empirically we find that while it generally works quite well if some  $P^* \in \mathcal{M}$  generates the data, it may sometimes fail if this is not the case.

## 4.10 Discussion

MDL is a versatile method for inductive inference: it can be interpreted in at least four different ways, all of which indicate that it does something reasonable. It is typically asymptotically consistent, achieving good rates of convergence. It achieves all this without having been designed for consistency, being based on a philosophy which makes no metaphysical assumptions about the existence of 'true' distributions. These facts strongly suggest that it is a good method to use in practice. Practical evidence confirms this in many contexts, in other contexts its behavior can be problematic. The main challenge for the future is to improve MDL for such cases, by somehow extending and further refining MDL procedures in a non ad-hoc manner. There is confidence that this can be done, and that MDL will continue to play an important role in the development of statistical, and more generally, in inductive inference.

**Further reading** Good places to start further exploration of MDL are [7] and [52]. Both papers provide excellent introductions, but they are geared towards a more specialized audience of information theorists and statisticians, respectively. Also worth reading is Rissanen's [93] monograph. While outdated as an introduction to MDL methods, this famous 'little green book' still serves as a great introduction to Rissanen's radical but appealing philosophy, which is described very eloquently.

<sup>†</sup> However, see [117] where it is pointed out that the problem of [64] disappears if a more reasonable coding scheme is used.

## 5. Model uncertainty

We consider the problems of variable selection and accounting for model uncertainty in linear models. Conditioning in a single selected model, ignores model uncertainty and thus leads to underestimation of uncertainty when making inferences about quantities of interest. The complete Bayesian solution to this problem involves averaging over all possible models when making inferences. This approach is often not practical. In this chapter we offer two alternative approaches. First, we describe a Bayesian model selection algorithm called ‘Occam’s window’ which involves averaging over a reduced set of models. Second, we describe a Markov chain Monte Carlo approach which directly approximates the exact solution.

The selection of subsets of predictor variables is a basic part of building a linear model. The objective of variable selection is typically stated as follows: given an independent variable  $Y$  and a set of candidate predictors  $X_1, X_2, \dots, X_k$ , find the best model of the form

$$Y = \beta_0 + \sum_{j=1}^p \beta_j X_j + \epsilon,$$

where  $X_1, X_2, \dots, X_p$  is a subset of  $X_1, X_2, \dots, X_k$ .

In this chapter we embed this model selection problem in the larger framework of accounting for model uncertainty. We argue that conditioning on a single selected model ignores model uncertainty, and that this, in turn, leads to underestimation of uncertainty when making inferences about quantities of interest. A complete Bayesian solution to this problem involves averaging over all possible models when making inferences about quantities of interest. Indeed, this approach provides optimal predictive ability [75]. In many applications however, this averaging will not be a practical proposition and here we present two alternative approaches. First, we extend the Bayesian model selection algorithm [75] to linear regression models. We refer to this algorithm as Occam’s window. Appealing to scientific norms, this approach involves averaging over a reduced set of models and allows for effective communication to the analyst. Second, we directly approximate the complete solution by applying the Markov chain Monte Carlo approach [76] to linear regression models. In this approach the posterior distribution of a quantity of interest is approximated by a Markov chain Monte Carlo method which generates a process that moves through model space.

### 5.1 Accounting for model uncertainty

A typical approach to data analysis is to carry out a model selection exercise leading to a single ‘best’ model and then to make inference as if the selected model were the true model. However, this ignores a major component of uncertainty, namely uncertainty about the model itself [28]. As a consequence, uncertainty about quantities of interest can be underestimated. For striking example of this see [76].

There is a standard Bayesian solution to this problem. If  $\mathcal{M} = \{M_1, \dots, M_K\}$  denotes the set of all models being considered and if  $\Delta$  is the quantity of interest such as a future observation or the utility of a course of action, then the posterior distribution of  $\Delta$  given the data  $D$  is

$$pr(\Delta|D) = \sum_{k=1}^K pr(\Delta|M_k, D)pr(M_k|D). \quad (5.1)$$

This is an average of the posterior distributions under each model weighted by the corresponding posterior model probabilities. In (5.1), the posterior distribution of model  $M_k$  is given by

$$pr(M_k|D) = \frac{pr(D|M_k)pr(M_k)}{\sum_{l=1}^K pr(D|M_l)pr(M_l)}, \quad (5.2)$$

where

$$pr(D|M_k) = \int pr(D|\theta_k, M_k)pr(\theta_k|M_k)d\theta_k, \quad (5.3)$$

is the marginal likelihood of model  $M_k$ ,  $\theta_k$  is the vector of parameters of model  $M_k$ ,  $pr(\theta_k|M_k)$  is the prior density of  $\theta_k$  under model  $M_k$ ,  $pr(D|\theta_k, M_k)$  is the likelihood, and  $pr(M_k)$  is the prior probability that  $M_k$  is the true model. All probabilities are implicitly conditional on  $\mathcal{M}$ , the set of all models being considered.

Implementations of the above strategy is difficult for two reasons. First, the integral in (5.3) can be hard to compute. Second, the number of terms in (5.1) can be enormous. In what follows we present workable solutions to both of these problems.

## 5.2 Bayesian framework and selection of prior distributions

Each model we consider is of the form

$$Y = \beta_0 + \sum_{j=1}^p \beta_j X_j + \epsilon = X\beta + \epsilon,$$

where the observed data on the predictors are contained in the  $n \times (p+1)$  matrix  $X$  and the observed data on the dependent variable are contained in the  $n$ -vector  $Y$ . We assign to  $\epsilon$  a normal distribution with zero mean and variance  $\sigma^2$  and assume that the  $\epsilon$ 's in distinct cases are independent. We consider the  $(p+1)$  parameters  $\beta$  and  $\sigma^2$  to be unknown.

Where possible, informative prior distributions for  $\beta$  and  $\sigma^2$  should be elicited and incorporated into the analysis [38]. In the absence of expert opinion we seek to choose prior distributions which reflect uncertainty about the parameters and also embody reasonable *a priori* constraints. We use the standard normal-gamma conjugate class,  $\mathcal{N}$ , of priors

$$\beta \approx \mathcal{N}(\mu, \sigma^2 V), \quad \frac{\nu\lambda}{\sigma^2} \approx \chi_\nu^2.$$

Here  $\nu, \lambda$ , the  $(p+1) \times (p+1)$  matrix  $V$  and  $(p+1)$ -vector  $\mu$  are hyperparameters to be chosen.

For non-categorical predictor variables we assume the individual  $\beta$ 's to be independent *a priori*. We center the distribution of  $\beta$  on zero (apart from  $\beta_0$ ) and choose  $\mu = (\hat{\beta}_0, 0, 0, \dots, 0)$  where  $\hat{\beta}_0$  is the ordinary least squares estimate of  $\beta_0$ . The covariance matrix  $V$  is diagonal with entries  $(s_y^2, \phi^2 s_1^{-2}, \phi^2 s_2^{-2}, \dots, \phi s_p^{-2})$  where  $s_y^2$  denotes the sample variance of  $Y$ ,  $s_i^{-2}$  denotes the sample variance of  $X_i$  for  $i = 1, \dots, p$ , and  $\phi$  is a hyperparameter to be chosen. The prior variance of  $\beta_0$  is chosen conservatively and represents an upper bound on the reasonable variance of this parameter. The variance of the remaining  $\beta$ -parameters are chosen to reflect increasing precision about each  $\beta_i$  as the variance of the corresponding  $X_i$  increases and to be invariant to scale changes in both the predictor variables and the response variable.

For a categorical predictor variable  $X_i$  with  $(c + 1)$  possible outcomes ( $c \geq 2$ ), the Bayes factors should be invariant to the selection of the corresponding dummy variables  $(X_{i1}, \dots, X_{ic})$ . To this end we set the prior variance of  $(\beta_{i1}, \dots, \beta_{ic})$  equal to  $\sigma^2 \phi^2 (\frac{1}{n} (X^i)^T X^i)^{-1}$  where  $X^i$  is the design matrix for the dummy variables, where each dummy variable has been centered by subtracting its sample mean. This is related to the  $g$ -prior in [126] and the complete prior covariance matrix for  $\beta$  is now given by

$$V(\beta) = \sigma^2 \begin{pmatrix} s_y^2 & & & & & \\ & \phi^2 s_1^{-2} & & & & \\ & & \ddots & & & \\ & & & \phi^2 s_{i-1}^{-2} & & \\ & & & & \phi^2 (\frac{1}{n} (X^i)^T X^i)^{-1} & \\ & & & & & \phi s_{i+1}^{-2} \\ & & & & & & \ddots \\ & & & & & & & \phi s_p^{-2} \end{pmatrix}$$

To choose the remaining hyperparameters  $\nu$ ,  $\lambda$  and  $\phi$ , we define a number of reasonable desiderata and attempt to satisfy them. In what follows we assume that all the variables have been standardized to have zero mean and sample variance one. We would like

- The prior density  $p(\beta_1, \dots, \beta_p)$  to be reasonable flat over the unit hypercube  $[-1, 1]^p$ .
- $p(\sigma^2)$  to be reasonably flat over  $(a, 1)$  for some small  $a$ .
- $pr(\sigma^2 \leq 1)$  to be large.

The order of importance of these desiderata is roughly the order in which they are listed. More formally, we maximize  $pr(\sigma \leq 1)$  subject to

- $$\frac{pr(\beta_1 = 0, \dots, \beta_p = 0)}{pr(\beta_1 = 1, \dots, \beta_p = 1)} \leq K_1,$$

following [59] we choose  $K_1 = \sqrt{10}$ ,

- $$\frac{\max_{a < \sigma^2 < 1} pr(\sigma^2)}{pr(\sigma^2 = a)} \leq K_2, \text{ and}$$

- $$\frac{\max_{a < \sigma^2 < 1} pr(\sigma^2)}{pr(\sigma^2 = 1)} \leq K_2.$$

Since desideratum 2 is less important than desideratum 1, we have chosen  $K_2 = 10$ .

For  $a = 0.05$  this yields  $\nu = 2.58$ ,  $\lambda = 0.28$ , and  $\phi = 2.85$ . For this set of hyperparameters  $pr(\sigma^2 \leq 1) = 0.81$ .

The marginal likelihood for  $Y$  under a model  $M_i$  based on the proper priors described above is given by

$$p(Y|\mu_i, V_i, X_i, M_i) = \frac{\Gamma(\frac{\nu+n}{2})(\nu\lambda)^{\frac{\nu}{2}}}{\pi^{\frac{n}{2}} \Gamma(\frac{\nu}{2}) |I + X_i V_i X_i^T|^{\frac{1}{2}}} \times (\lambda\nu + (Y - X_i \mu_i)^T (I + X_i V_i X_i^T)^{-1} (Y - X_i \mu_i))^{-\frac{\nu+n}{2}}, \quad (5.4)$$

where  $X_i$  is the design matrix and  $V_i$  is the covariance matrix for  $\beta$  corresponding to model [85]. The Bayes factor for  $M_0$  versus  $M_1$ , the ratio of (5.4) for  $i = 0$  and  $i = 1$ , is then given by

$$B_{01} = \left( \frac{|I + X_1 V_1 X_1^T|}{|I + X_0 V_0 X_0^T|} \right)^{\frac{1}{2}} \times \left( \frac{\lambda\nu + (Y - X_0 \mu_0)^T (I + X_0 V_0 X_0^T)^{-1} (Y - X_0 \mu_0)}{\lambda\nu + (Y - X_1 \mu_1)^T (I + X_1 V_1 X_1^T)^{-1} (Y - X_1 \mu_1)} \right)^{-\frac{\nu+n}{2}}. \quad (5.5)$$

### 5.3 Model selection using Occam's window

Our first way of accounting for model uncertainty starting from (5.1) involves applying the Occam's window algorithm [75] to linear regression models. Two basic principles underly this approach. First, if a model predicts the data far less well than the model which provides the best predictors, then it has effectively been discredited and should no longer be considered. Thus the model not belonging to

$$\mathcal{A}' = \left\{ M_k : \frac{\max_l (pr(M_l|D))}{pr(M_k|D)} \leq C \right\}, \quad (5.6)$$

should be excluded from (5.1) where  $C$  is chosen by the data analyst. Second, appealing to Occam's razor, we exclude models which receive less support from the data than any of their simpler submodels. More formally we also exclude from (5.1) models belonging to

$$\mathcal{B} = \left\{ M_k : \exists M_l \in \mathcal{A}, M_l \subset M_k, \frac{pr(M_l|D)}{pr(M_k|D)} > 1 \right\}. \quad (5.7)$$

Equation (5.1) is then replaced by

$$pr(\Delta|D) = \frac{\sum_{M_k \in \mathcal{A}} pr(\Delta|M_k, D) pr(D|M_k) pr(M_k)}{\sum_{M_k \in \mathcal{A}} pr(D|M_k) pr(M_k)}, \quad (5.8)$$

where

$$\mathcal{A} = \mathcal{A}' \setminus \mathcal{B}. \quad (5.9)$$

This greatly reduces the number of models in the sum in (5.1) and now all that is required is a search strategy. First, if a model is rejected then all its submodels are rejected. The second principle - 'Occam's window' - concerns the interpretation of

the ratio of posterior model probabilities  $pr(M_1|D)/pr(M_0|D)$ . Here model  $M_0$  is a model with one less predictor than  $M_1$ . The essential idea is shown in figure 5.1. If there is evidence for  $M_0$  then  $M_1$  is rejected, but to reject  $M_0$  we require strong evidence for the larger model,  $M_1$ . If the evidence is inconclusive (falling in Occam's window) neither model is rejected.

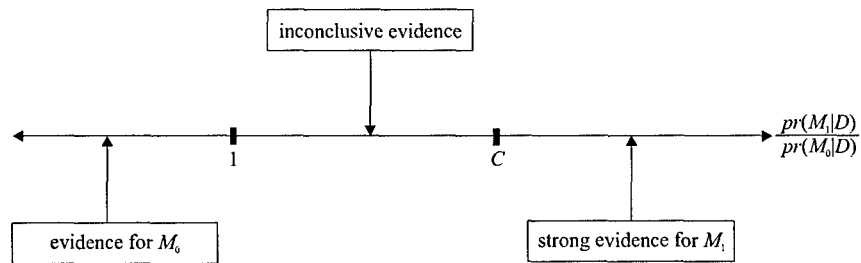


Figure 5.1: Occam's window, interpreting the posterior odds for nested models.

These principles define the strategy. Typically, the number of terms in (5.1) is reduced to fewer than 25, and often to as few as one or two. A description of the algorithm is given below.

The search can proceed in two directions: 'Up' from each starting model by adding variables, or 'Down' from each starting model by dropping variables. When starting from a model made up of some subset of variables, we first execute the 'Down' algorithm. Then we execute the 'Up' algorithm, using models from the 'Down' algorithm as a starting point. Experience to date suggests that the ordering of these operations has little impact on the final set of models. Let  $\mathcal{A}$  and  $\mathcal{C}$  be subsets of model space  $\mathcal{M}$ , where  $\mathcal{A}$  denotes the set of 'acceptable' models and  $\mathcal{C}$  denotes the models under consideration. For both algorithms we begin with  $\mathcal{A} = \emptyset$  and  $\mathcal{C} =$  set of starting models.

- Down algorithm

1. Select a model  $M$  from  $\mathcal{C}$
2.  $\mathcal{C} \leftarrow \mathcal{C} - \{M\}$  and  $\mathcal{A} \leftarrow \mathcal{A} + \{M\}$
3. Select a submodel  $M_0$  of  $M$  by removing a variable from  $M$
4. Compute

$$B = \log \left( \frac{pr(M_0|D)}{pr(M|D)} \right)$$

5. If  $B > O_R$  then  $\mathcal{A} \leftarrow \mathcal{A} - \{M\}$  and if  $M_0 \notin \mathcal{C}$ ,  $\mathcal{C} \leftarrow \mathcal{C} + \{M_0\}$
6. If  $O_L \leq B \leq O_R$  then if  $M_0 \notin \mathcal{C}$ ,  $\mathcal{C} \leftarrow \mathcal{C} + \{M_0\}$
7. If there are more submodels of  $M$ , go to 3
8. If  $\mathcal{C} \neq \emptyset$ , go to 1

- Up algorithm

1. Select a model  $M$  from  $\mathcal{C}$
2.  $\mathcal{C} \leftarrow \mathcal{C} - \{M\}$  and  $\mathcal{A} \leftarrow \mathcal{A} + \{M\}$
3. Select a supermodel  $M_1$  of  $M$  by adding a variable to  $M$
4. Compute

$$B = \log \left( \frac{pr(M|D)}{pr(M_1|D)} \right)$$

5. If  $B < O_L$  then  $\mathcal{A} \leftarrow \mathcal{A} - \{M\}$  and if  $M_1 \notin \mathcal{C}$ ,  $\mathcal{C} \leftarrow \mathcal{C} + \{M_1\}$
6. If  $O_L \leq B \leq O_R$  then if  $M_1 \notin \mathcal{C}$ ,  $\mathcal{C} \leftarrow \mathcal{C} + \{M_1\}$
7. If there are more supermodels of  $M$ , go to 3
8. If  $\mathcal{C} \neq \emptyset$ , go to 1

Upon termination,  $\mathcal{A}$  contains the set of potentially acceptable models. Finally, we remove all the models which satisfy (5.7), where 1 is replaced by  $\exp(O_R)$ , and those models for which

$$\frac{\max_l(\text{pr}(M_l|D))}{\text{pr}(M_k|D)} > C.$$

$\mathcal{A}$  now contains the acceptable models.

## 5.4 Markov chain Monte Carlo model composition

Our second approach is to approximate (5.1) using the Markov chain Monte Carlo model composition approach [76]. This generates a stochastic process which moves through model space. Specifically, let  $\mathcal{M}$  denote the space of models under consideration. We can construct a Markov chain  $\{M(t), t = 1, 2, \dots\}$  with state space  $\mathcal{M}$  and equilibrium distribution  $\text{pr}(M_i|D)$ . If we simulate this Markov chain for  $t = 1, \dots, N$ , then under certain regularity conditions, for any function  $g(M_i)$  defined on  $\mathcal{M}$ , the average

$$\hat{G} = \frac{1}{N} \sum_{t=1}^N g(M(t)), \quad (5.10)$$

is a simulation - consistent estimate of  $E(g(M))$  [105]. To compute (5.1) in this fashion set  $g(M) = \text{pr}(\Delta|M, D)$ .

To construct the Markov chain we define a neighborhood  $\text{nbd}(M)$  for each  $M \in \mathcal{M}$  which consists of the model  $M$  itself and the set of models with either one variable more or one variable fewer than  $M$ . Define a transition matrix  $q$  by setting  $q(M \rightarrow M') = 0$  for all  $M' \notin \text{nbd}(M)$  and  $q(M \rightarrow M')$  constant for all  $M' \in \text{nbd}(M)$ . If the chain is currently in state  $M$ , we proceed by drawing  $M'$  from  $q(M \leftarrow M')$ . It is then accepted with probability

$$\min \left( 1, \frac{\text{pr}(M'|D)}{\text{pr}(M|D)} \right).$$

Otherwise the state stays in  $M$ .

## 5.5 Freedman's paradox resolved

Linear regression models are frequently used even when little is known about the relationship between the predictors and the response. In [35] it was shown that in the extreme case where there is no relationship between the predictors and the response variable, omitting the predictors with the smallest  $t$ -values (e.g,  $p > 0.25$ ) can result in a model with a highly significant  $F$  statistic and high  $R^2$ . We will refer to this unfortunate phenomenon as 'Freedman's paradox'. In contrast, if the response and predictors



are independent, Occam's window typically indicates the null model only, or as one of a small number of 'best' models.

As in [35], we generated 5100 independent observations from a standard normal distribution to create a matrix with 100 rows and 51 columns. The first column was taken to be the dependent variable in a regression equation and the other 50 columns were taken to be the predictors. Thus the predictors are independent of the response by construction. For the entire data set, the multiple regression results were as follows

- $R^2 = 0.55, p = 0.29,$
- 18 coefficients out of 50 were significant at the 0.25 level, and
- 4 coefficients out of 50 were significant at the 0.05 level.

Three different variable selection procedures were used on the simulated data. The first of these was the method used in [35]. Here all predictors with  $p$ -values of 0.25 or lower were included in a second pass over the data. The results of this method were as follows

- $R^2 = 0.40, p = 0.0003,$
- 17 coefficients out of 18 were significant at the 0.25 level, and
- 10 coefficients out of 18 were significant at the 0.05 level.

These results are highly misleading as they indicate a definite relationship between the response and the predictors, whereas, in fact, the data are all noise.

The second model selection method used on the full data set was Efroymson's stepwise method [78]. This method indicated a model with 15 predictors with the following results

- $R^2 = 0.40, p = 0.0001,$
- all 15 were significant at the 0.25 level, and
- 10 coefficients out of 15 were significant at the 0.05 level.

Again a model is chosen which misleadingly appears to have a great deal of explanatory power.

The third variable selection method was Occam's window. The only model chosen by this method was the null model. The procedure described above was repeated 10 times with similar results. In 5 simulations, Occam's window chose only the null model. For the remaining simulations 3 models or fewer were chosen along with the null model. For the non-null models that were chosen, all models had  $R^2$  values less than 0.15. For all of the simulations the selection procedure used in [35] and Efroymson's stepwise method chose models with many predictors and highly significant  $R^2$  values.

*Table 5.1: Log predictive scores for the Freedman simulated noise data.*

Model	Method	log predictive score
null model	Occam's window	133
18 predictors	Freedman [35]	174
15 predictors	Efroymson	181

To compare the predictive performance of the models chosen by the three methods, another data set with 100 rows and 51 columns was simulated and log predictive scores were calculated (see table 5.1)

The measure of predictive ability is the logarithm scoring rule described in [42] which is based on the conditional predictive ordinate [39]. Specifically, we measured the predictive ability of an individual model,  $M$ , with

$$- \sum_{d \in D \setminus D^T} \log(pr(d|M, D^T)).$$

We measured the predictive performance for model averaging with

$$- \sum_{d \in D \setminus D^T} \log \left( \sum_{M \in \mathcal{A}} pr(d|M, D^T) pr(M|D^T) \right),$$

where for Occam's window  $\mathcal{A}$  is the set of selected models and for Markov chain Monte Carlo model composition  $\mathcal{A}$  is the set of visited models. The log predictive score for the only model selected by Occam's window (the null model) is considerably better than the log predictive score for the models chosen by the other two methods. In addition, the mean square predictive error was calculated. The mean square predictive error for Freedman's method was 1.4 and the mean square predictive error for the Efroymsen model was 1.5 while the mean square predictive error for the null model was 0.9. Thus Occam's window has considerably greater out-of-sample predictive power than the more standard variable selection methods considered.

At best, Occam's window correctly indicates that the null model is the only model that should be chosen when there is no signal in the data. At worst, Occam's window chooses the null model along with several other models. The presence of the null model among those chosen by Occam's window should indicate to a researcher that there may be evidence for a lack of signal in the analyzed data. Thus Occam's window largely resolves 'Freedman's paradox'.

## 5.6 Discussion

In [28] the problem of assessing model uncertainty has also been addressed. This approach was based on the idea of model expansion, i.e., starting with a single reasonable model chosen by a data-analytic search, expanding model space to include those models which are suggested by context or other considerations, and then averaging over this model class. In [28] the problem of model uncertainty in variable selection is not addressed directly. However, one could consider Occam's window to be a practical implementation of model expansion.

In [41] the stochastic search variable selection method similar in spirit to Markov chain Monte Carlo model composition has been developed. Here a Markov chain is defined which moves through model space and parameter space at the same time. To make the chain irreducible, however, their method never actually removes a predictor from the full model, but only sets it close to zero with high probability. If this probability is very high, the algorithm has convergence difficulties, and if not the results can be hard to interpret. Our new approach avoids this problem by integrating analytically over parameter space.

The prior distribution for the covariance matrix for  $\beta$  depends on the actual data, including both the dependent and independent variables. A similar data dependent approach to the assessment of the priors was used in [84]. While this may appear at first sight to be contrary to the idea of a prior, our objective was to develop priors that lead to posteriors similar to those of a person with little prior information. Examples analyzed to date suggest that this objective was achieved. The priors for  $\beta$  lead to a reasonable prior variance and result in conclusions that are not highly sensitive to the choice of hyperparameters. Thus the data dependence does not appear to be a drawback.

The choice of which procedure to use - Occam's window or Markov chain Monte Carlo model composition - will depend on the particular application. Occam's window will be most useful when one is interested in making inferences about the relationship between variables. Occam's window also tends to be much faster computationally. Markov chain Monte Carlo model composition is the better procedure to choose when the goal is good predictions or if the posterior distribution of some quantity is of more interest than the nature of the 'true' model and if computer time is not a critical consideration. However, each approach is flexible enough to be used successfully for both inference and prediction.



## 6. Making predictions reliable

According to the MDL principle, models of data are always probabilistic; if a class of non-probabilistic models is used to model the data at hand, it is first mapped to a corresponding probabilistic class. On the other hand, these models are to be interpreted as codes for the data - not as traditional probability distributions according to which the data are generated. This raises the question of what conclusions (predictions) about future data can and what conclusions cannot be drawn on the basis of such 'probabilistic' models. The question becomes all the more difficult if we acknowledge, in line with the MDL philosophy, that our models will always be partially wrong - even if they allow us to substantially compress the data.

In this chapter, we identify conditions under which a probabilistic model, inferred from the data, can be used to reliably predict future data even if that model is really a probabilistic representation of a non-probabilistic model and/or if the model is wrong. We show that given a model class  $\mathcal{M}$  with a fixed number of parameters ( $\mathcal{M}$  is not necessarily probabilistic) and an error function ER, we can turn  $\mathcal{M}$  into a probabilistic version  $\langle \mathcal{M} \rangle_{\text{ER}}$  that is essentially equivalent to  $\mathcal{M}$  except that it leads to 'reliable' estimates of the error function. We call  $\langle \mathcal{M} \rangle_{\text{ER}}$  the entropification of  $\mathcal{M}$ . Entropification stands at the basis of the main results of this chapter (theorems 6.1 - 6.3) which can be summarized as follows:

1. Under the assumption that the data are i.i.d. according to an essentially arbitrary unknown 'true' probability distribution  $P^*$ , we can infer from a large enough data set  $D$ , with high probability, a model  $\tilde{\theta}$  in  $\langle \mathcal{M} \rangle_{\text{ER}}$  that is
  - (a) the optimal model in  $\langle \mathcal{M} \rangle_{\text{ER}}$  for predicting future data against error function ER. Among the models in  $\langle \mathcal{M} \rangle_{\text{ER}}$ ,  $\tilde{\theta}$  minimizes the 'true' expected error  $E_{P^*}(\text{ER}(Y|\tilde{\theta}, X))$ , and
  - (b) can be 'reliably' used, since it gives a truthful impression of its own performance in the sense that  $E_{\tilde{\theta}}(\text{ER}(Y|\tilde{\theta}, X)) = E_{P^*}(\text{ER}(Y|\tilde{\theta}, X))$ .

Essentially, this means that whenever the assumption that the data are i.i.d. can be justified and the function ER according to which errors will be measured is known, the model  $\tilde{\theta}$  can be used (1) to arrive at optimal predictions (relative to the model class  $\langle \mathcal{M} \rangle_{\text{ER}}$ ) of future data against ER and (2) as an accurate estimator of how good predictions will be - even if  $\mathcal{M}$  is a wrong ('misspecified') model class that does not contain any model that is similar to the 'true'  $P^*$ .

While  $\tilde{\theta}$  can be inferred from data by many statistical inference procedures (not necessarily MDL), the 'entropification' of  $\mathcal{M}$  turns out to yield additional results when combined with MDL, leading to the other two important results of this chapter

2. Entropification removes an inherent arbitrariness in MDL's trade-off between error and model complexity that occurs when non-probabilistic model classes are used (section 6.2), and
3. Entropification allows us to associate codes with non-probabilistic model classes in an optimal manner, in the sense that the 'worst case expected code length' is minimized (proposition 6.8).

It is well-known that the modeling error using the normal distribution with varying  $\sigma^2$  works even when the errors are not truly normally distributed [13], so this far, there is nothing new here. Our own contribution lies in the fact that we consider the general case of (almost) arbitrary error functions ER and model classes  $\mathcal{M}$ . We will give a recipe of how, given  $\mathcal{M}$  and ER, one can define a new probabilistic class  $\langle \mathcal{M} \rangle_{\text{ER}}$  that has some special properties. We call the model class  $\langle \mathcal{M} \rangle_{\text{ER}}$  the entropification of  $\mathcal{M}$  with respect to ER.  $\langle \mathcal{M} \rangle_{\text{ER}}$  is constructed from  $\mathcal{M}$  by adding a single extra real-value parameter  $\beta$  as part of the hypotheses: models in  $\langle \mathcal{M} \rangle_{\text{ER}}$  are indexed by parameters  $\theta = (H, \beta)$  for some  $H \in \mathcal{M}$  and  $\beta \in \mathbb{R}$ . If  $(H, \beta)$  is inferred from data  $D$ , then the  $\beta$  associated with  $H$  can be interpreted as a reliable estimate of the error  $H$  will make on future data.  $\beta$  will also determine the entropy of the model  $(H, \beta)$ , hence the name ‘entropification’. We give three examples corresponding to three often used error functions.

- If  $\mathcal{M}$  is a class of continuous functions and ER is the squared error, then  $\langle \mathcal{M} \rangle_{\text{ER}}$  turns out to be equivalent to the class  $\{P(\cdot|H, \sigma^2, \cdot) | H \in \mathcal{M}; \sigma^2 > 0\}$  where

$$P(y|H, x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - H(x))^2}{2\sigma^2}\right). \quad (6.1)$$

If  $(H, \sigma^2)$  is inferred from  $D$ , then  $\sigma^2$  can be interpreted as an estimate of the squared error  $H$  will make on future data.

- Let  $\mathbb{E}$  be a sample space. If  $\mathcal{M}$  is a class of concepts (functions mapping  $\mathbb{E}_x$  on  $\mathbb{E}_y = \{0, 1\}$ ) and ER is the 0/1-error, defined by

$$\text{ER}_{01}(y|H, x) = \begin{cases} 0 & \text{if } H(x) = y \\ 1 & \text{otherwise} \end{cases}, \quad (6.2)$$

then  $\langle \mathcal{M} \rangle_{\text{ER}}$  is equivalent to a class of distributions  $\{P(\cdot|H, \theta, \cdot) | H \in \mathcal{M}; 0 < \theta < 1\}$  where

$$E_{P(\cdot|H, \theta, \cdot)}(\text{ER}(Y|H, X)) = \theta.$$

$\theta$  can be interpreted as the probability that  $H(X) \neq Y$ . If  $(H, \theta)$  is inferred from  $D$ , then  $\theta$  can be interpreted as an estimate of the 0/1-error that  $H$  will make on future data.

- Let  $\mathcal{M}$  be a model class that is finitely parameterized by some  $\Gamma$ . If  $\mathcal{M}$  is a class of probabilistic models  $\{P(\cdot|\eta) | \eta \in \Gamma\}$  and ER is the logarithmic error, then  $\langle \mathcal{M} \rangle_{\text{ER}}$  will turn out to be equivalent to a class  $\{P(\cdot|\eta, \beta) | \eta \in \Gamma; \beta \in \mathbb{R}\}$ , where

$$P(y|\eta, \beta) = \frac{1}{Z_\eta(\beta)} \exp(\beta \ln(P(y|\eta))) = \frac{P(y|\eta)^\beta}{\sum_{y \in \mathbb{E}_y} P(y|\eta)^\beta}.$$

If  $(\eta, \beta)$  is inferred from  $D$ , then  $\beta$  can be interpreted as an estimate of the logarithmic error that  $\eta$  will make on future data.

This chapter is organized as follows. In section 6.1 we formally introduce the concept of ‘entropification’, present some of its basic properties and give some examples of its use and present our main results (item 1 in the listing above). In section 6.2 we show how entropification can be used in the context of MDL.

## 6.1 Entropification of a model class

In this section we formally introduce the concept of ‘entropification’ and give some examples of its use and present our main results. We will assume that we use some reasonable inference procedure that, for those sequences  $x_1, x_2, \dots$ , there really is ‘something to infer’, and that it is guaranteed to work for large enough samples.

### Definition 6.1

Let  $\mathcal{M}$  be a model class that is finitely parameterized by some  $\Gamma$ . Let  $\mathcal{L}_{\mathcal{M}}$  be an estimation procedure that, for each  $n$ ,  $x^n \in \mathbb{E}^n$  outputs an estimator  $\tilde{\theta}(x^n) \in \Gamma$ . We call  $\mathcal{L}_{\mathcal{M}}$  reasonable if for every sequence  $x_1, x_2, \dots$  for which the maximum likelihood estimate  $\hat{\theta}(x^n)$  converges to some value,  $\tilde{\theta}(x^n)$  converges to that same value; that is

if  $\lim_{n \rightarrow \infty} \hat{\theta}(x^n)$  exists and is equal to  $\hat{\theta}_{fut} \in \Gamma$   
then  $\lim_{n \rightarrow \infty} \tilde{\theta}(x^n)$  must also exist and be equal to  $\hat{\theta}_{fut}$ .

Apparently, some aspects of the data can be reliably predicted on the basis of the maximum entropy model while others cannot. We will now define the notions of reliable estimations and decisions. The following definition captures the idea of reliable estimation. In the definition,  $\text{int}(\mathbb{U})$  stands for the interior of the set  $\mathbb{U}$ .

### Definition 6.2

Let  $\mathcal{M}$  be a class of probabilistic models over  $\mathbb{E}$  parameterized by some  $\Gamma$ . Let  $\psi : \mathbb{E} \rightarrow \mathbb{U}$  be a given function. If, for all  $n$ , all  $x^n \in \mathbb{E}^n$  with  $\hat{\theta}(x^n) \in \text{int}(\Gamma)$ , we have

$$E_{\hat{\theta}(x^n)}(\psi(X)) = \overline{\psi(x)}^n, \quad (6.3)$$

and, moreover,  $E_{\hat{\theta}}(\psi(X))$  is a continuous function of  $\theta$ , then we say that averages of  $\psi$  can be reliably estimated on the basis of  $\mathcal{M}$ . Otherwise we say that averages of  $\psi$  cannot be reliably estimated on the basis on  $\mathcal{M}$ .

We only consider the case where  $\mathcal{M}$  can be parameterized by a fixed number of parameters  $k$ . This is formalized in the following definition:

### Definition 6.3

Let  $\mathcal{M}$  be a class of probabilistic models over sample space  $\mathbb{E}$  and let  $\Gamma \subset \mathbb{R}^k$ . We say that  $\mathcal{M}$  is finitely parameterized by  $\Gamma$  if

1. there exists a bijection  $g : \Gamma \rightarrow \mathcal{M}$ ,
2.  $D \in \mathbb{E}^*$  is arbitrary but fixed, then  $P(D|\theta)$  as a function of  $\theta$  is the restriction to domain  $\Gamma$  of a continuous function  $\psi : \mathbb{R}^k \rightarrow \mathbb{R}$ ,
3.  $\mathbb{E}$  is continuous, then for all  $n$ , the density function  $f(x^n|\theta)$  as a function of  $x^n$  is continuous at each  $x^n$  in the interior of  $\mathbb{E}^n$ .

Throughout the remainder of this chapter we assume that all error functions ER considered are sufficiently regular. More precisely, let  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$ , and let  $\text{ER} : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{R}$  be an error function defined as follows

**Definition 6.4**

Let  $\mathcal{M}$  be a set and  $\mathbb{E}$  be a sample space. An error function for  $\mathcal{M}$  is a total function  $\text{ER} : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{U}$  where  $\mathbb{U} \subseteq \mathbb{R}$ . For  $x \in \mathbb{E}$  and  $H \in \mathcal{M}$ , we write  $\text{ER}(X|H)$  rather than  $\text{ER}(x, H)$ . We restrict ourselves to additive error functions:  $\text{ER}$  is extended to outcomes  $x^n \in \mathbb{E}^n$  by  $\text{ER}(x^n|H) = \sum_{i=1}^n \text{ER}(x_i|H)$ .

Our assumption is that, for all fixed  $H \in \mathcal{M}$ ,  $\text{ER}(y|H, x)$  considered as a function of  $x$  and  $y$  can be used to define a maximum entropy model class. Let  $\phi = (\phi_1, \dots, \phi_m)$  be a function with domain  $\mathbb{E}$  and range  $\mathbb{U} = \mathbb{U}_1 \times \dots \times \mathbb{U}_m$ , and let  $t = (t_1, \dots, t_m) \in \mathbb{U}$ . We require the constraint  $E(\phi(X)) = t$  to be such that for  $i = 1, \dots, m$ . Specifically, we assume that for  $H \in \mathcal{M}$ , the function  $\phi_H$  over  $\mathbb{E}_x \times \mathbb{E}_y$  defined by  $\phi_H(x, y) \equiv \text{ER}(y|H, x)$  satisfies the following conditions:

- C1  $\mathbb{U}_i$  is the smallest interval in  $\mathbb{R}$  such that  $\forall x \in \mathbb{E}: \phi(x) \in \mathbb{U}_i$ ,
- C2 If  $\mathbb{E}$  is continuous, then  $\phi_i$  is continuous. More precisely, if  $\mathbb{E} \subseteq \mathbb{R}^k$ , then  $\phi_i$  is the restriction to domain  $\mathbb{E}$  of some continuous function  $\psi: \mathbb{R}^k \rightarrow \mathbb{R}$ ,
- C3 In the discrete case  $\mathbb{E}$  contains a finite number of elements. In the continuous case  $\mathbb{E}$  can be written as  $\mathbb{E}_1 \times \dots \times \mathbb{E}_l$  for some  $l \geq 1$ , where for each  $\mathbb{E}_j$  with  $1 \leq j \leq l$ 
  - 1.  $\mathbb{E}_j$  is a closed interval in  $\mathbb{R}$ , or
  - 2.  $\mathbb{E}_j = \mathbb{R}$  and there exist  $\alpha > 0$  and  $C \in \mathbb{R}$  such that

$$\forall x_1, \dots, x_l: \phi_i(x_1, \dots, x_j, \dots, x_l) \geq |x_j|^\alpha - C.$$

This ensures that the maximum entropy model class for  $\phi_H(x, y)$  exists. But we need something stronger than merely the guaranteed existence of this class, as we will now explain.

Throughout this chapter, we consider two cases. In the first case, the hypotheses class  $\mathcal{M}$  contains models relating to  $\mathbb{E}_y$  and not  $\mathbb{E}_x$  (for example, each  $H \in \mathcal{M}$  is itself a probabilistic model over  $\mathbb{E}_y$  or each  $H$  is a relation over  $\mathbb{E}_y$  not involving  $\mathbb{E}_x$ ). In such a case  $\mathbb{E}_x$  does not really play any role and we could have equally well set  $\mathbb{E} = \mathbb{E}_y$ . In this situation, the entropification of a model class  $\mathcal{M}$  with respect to error function  $\text{ER} : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{R}$  is the class of probabilistic models containing, for each  $H \in \mathcal{M}$ , the class of distributions  $P(\cdot|H, \beta)$  defined by

$$P(y|H, \beta) = \frac{1}{Z_H(\beta)} \exp(-\beta \text{ER}(y|H)), \quad (6.4)$$

where  $Z_H(\beta)$  is a normalizing factor

$$Z_H(\beta) \equiv \sum_{y \in \mathbb{E}_y} \exp(-\beta \text{ER}(y|H)), \quad (6.5)$$

and ranges over all  $\beta \in \mathbb{R}$  for which  $P(y|H, \beta)$  is well-defined. As can be seen, the distributions (6.4) are formally equivalent to maximum entropy distributions.

The formal definition of entropification, we give below, unifies the unconditional case with the more complicated conditional or supervised case. In the latter case, we assume that the outcomes  $x \in \mathbb{E}_x$  do play a role, and we are interested in the conditional version of (6.4),  $P(y|H, \beta, x) = Z_{H,x}^{-1}(\beta) \exp(-\beta \text{ER}(y|H, x))$  with  $Z_{H,x}(\beta) \equiv \sum_{y \in \mathbb{E}_y}$



$\exp(-\beta \text{ER}(y|H, x))$ . However, all our results will only hold if the resulting distributions are still ‘essentially’ maximum entropy distributions. For this reason, we must additionally assume that the following conditions hold

C4  $\mathbb{E}_x$  is either finite or compact,

C5 ER is such that for each fixed  $H$  and each fixed  $\beta \in \mathbb{R}$ ,  $Z_{H,x}(\beta)$  is either equal for all  $x \in \mathbb{E}_x$  or it diverges for all  $x \in \mathbb{E}_x$ .

C5 turns out to hold for most error functions ER of interest. These include error functions like the squared and 0/1-error. C5 allows us to drop the subscript  $x$  in  $Z_{H,x}(\beta)$  and write, for arbitrary  $x \in \mathbb{E}_x$

$$Z_H(\beta) = \sum_{y \in \mathbb{E}_y} \exp(-\beta \text{ER}(y|H, x)). \quad (6.6)$$

In the remainder of this chapter, we tacitly assume ER to satisfy conditions C1-C5, omitting the divergent case. We are now ready to define entropification formally.

#### Definition 6.5 (Entropification)

Let  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$ . The entropification of a model class  $\mathcal{M}$  with respect to error function  $\text{ER} : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{R}$  is the class of (conditional) probabilistic models

$$\langle \mathcal{M} \rangle_{\text{ER}} = \{P(\cdot|\theta, \cdot) | \theta = (H, \beta); H \in \mathcal{M}; \beta \in \Gamma_{\text{nat}}(H)\}. \quad (6.7)$$

Here  $P(\cdot|\theta, \cdot) = P(\cdot|H, \beta, \cdot)$  is a conditional model defined as follows:

1. For each  $x \in \mathbb{E}_x$ ,  $P(\cdot|\theta, x) = P(\cdot|H, \beta, x)$  is a probabilistic distribution over  $\mathbb{E}_y$  defined by

$$P(y|H, \beta, x) = \frac{1}{Z_H(\beta)} \exp(-\beta \text{ER}(y|H, x)) \quad \text{for all } y \in \mathbb{E}_y. \quad (6.8)$$

Here  $Z_H(\beta)$  is as in (6.6).

2. For all  $(x^n, y^n) \in \mathbb{E}^n$ ,  $P(y^n|H, \beta, x^n) = \prod_{i=1}^n P(y_i|H, \beta, x_i)$ .

For each  $H$ , the set of  $\beta$ , such that  $P(\cdot|H, \beta, \cdot) \in \langle \mathcal{M} \rangle_{\text{ER}}$ , is given by

$$\Gamma_{\text{nat}}(H) = \{\beta | Z_H(\beta) < \infty\},$$

where  $\Gamma_{\text{nat}}$  is the natural parameter space.

If  $\mathbb{E}$  is continuous, the sum in  $Z_H(\beta)$  gets replaced by the corresponding integral.  $Z_H(\beta)$  acts as normalizing constant.

The code corresponding to  $P(\cdot|H, \beta, \cdot)$  leads to the following code lengths (expressed in nats [20])

$$L(y^n|H, \beta, x^n) = -\ln(P(y^n|H, \beta, x^n)) = \beta \sum_{i=1}^n \text{ER}(y_i|H, x_i) + n \ln(Z_H(\beta)). \quad (6.9)$$

We see that the code length of  $y^n$  given  $H, \beta, x^n$  contains an error term and a ‘uniform’ term  $n \ln(Z_H(\beta))$  that grows linearly in  $n$  and is equal for all  $y^n$ . This shows that  $\beta$  can be interpreted as determining how strongly the error should be weighted in the code

length corresponding to hypotheses  $(H, \beta)$ . The extreme case  $\beta = 0$  corresponds, for each  $x_i$ , to the uniform distribution over all outcomes in  $\mathbb{E}_y$ . For fixed  $H$ , in the limit for  $\beta \rightarrow \infty$ , the probability under  $(H, \beta)$  of an outcome  $y^n$  given  $x^n$  with  $\text{ER}(y^n|H, x^n) > 0$  becomes 0.

### Example 6.1

Consider the fitting of polynomials. Let data  $D = (x^n, y^n)$  be given. In a non-probabilistic approach to this problem, we would use some algorithm that, for each  $D$ , when input  $D$ , outputs a polynomial  $\hat{H}$  that it regards as an optimal hypotheses for  $D$ . Such a polynomial  $\hat{H}$  in itself does not give any information on how good it will be on future data, and this can be problematic. For example, imagine that a company uses some sophisticated tool to infer  $\hat{H}$  from lots of data, and then sells  $\hat{H}$  to a client so that the client can use it to predict future data ( $\hat{H}$  may, for example, be a model for some data from the stock exchange and the client may use it as a guideline for future investments). If the company only gives  $\hat{H}$  to the client, then the client has no means of knowing how well  $\hat{H}$  actually will predict future data. This can be most easily demonstrated if we imagine that the model class  $\mathcal{M}$  is restricted to the class of first-degree polynomials. Let us denote  $D_1$  a first data set,  $D_2$  a second data set and  $\hat{H}$  an optimal first degree polynomial. Assuming that the company uses a reasonable (definition 6.1) method to infer the best polynomial, it will infer a polynomial reasonably close to  $\hat{H}$  for both data sets. However, if future data behaves like present data, then in the case of  $D_1$ ,  $\hat{H}$  will be a much better predictor than in the case of  $D_2$ . The client (who has not seen the ‘training’ data) would probably like to know how good the hypotheses  $\hat{H} \approx \hat{H}$  is before it decides whether to buy it or not; but  $\hat{H}$  does not reveal this information. Therefore, the client may rather want the company to sell a tuple  $(\hat{H}, \hat{\sigma}^2)$  where  $\hat{\sigma}^2$  is some reasonable estimate of the error  $\hat{H}$  will make on future data. In this way, he will get a reliable impression of the performance of  $\hat{H}$ . Now, let  $\mathcal{M}$  be a class of continuous functions  $H : \mathbb{E}_x \rightarrow \mathbb{E}_y$ . From the definition of entropification (definition 6.5) we can see (by substituting  $\beta = (\frac{1}{2}\sigma^2)$ ) that  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$ , the entropification of  $\mathcal{M}$  with respect to the squared error, is equivalent to the model class that supplies  $\mathcal{M}$  with a normal error distribution of arbitrary variance  $\sigma^2 > 0$ . Formally

$$\langle \mathcal{M} \rangle_{\text{ER}_{sq}} = \{P(\cdot|H, \sigma^2, \cdot) | H \in \mathcal{M}; \sigma^2 > 0\}, \quad (6.10)$$

with  $P(\cdot|H, \sigma^2, \cdot)$  as given by (6.1). In this case, the value of  $Z_H(\beta)$  is independent of  $H$  and finite for all  $\beta > 0$ . The parameter space will be  $\Gamma_{\text{nat}}(H) = \{\beta | \beta > 0\}$  independently of  $H$ , corresponding to all variances  $\sigma^2 = 1/(2\beta) > 0$ . Note that,  $E_{P(\cdot|H, \sigma^2, \cdot)}(\text{ER}_{sq}(Y|H, X)) = \sigma^2$ . Hence when the tuple  $(H, \beta)$  is inferred from data, then  $\beta$ , which determines  $\sigma^2$ , can be seen as an estimate of the expected squared error  $H$ .  $\diamond$

### Example 6.2 (Concept learning and Bernoulli parameters)

Let  $\mathcal{M}$  be a class of concepts over  $\mathbb{E} = \mathbb{E}_x \times \{0, 1\}$  and let  $\text{ER}_{01}$  be the 0/1-error function (see (6.2)). Let the observed data  $D = (x^n, y^n)$ . Let  $\langle \mathcal{M} \rangle_{\text{ER}_{01}}$  be the entropification of  $\mathcal{M}$  with respect to error function  $\text{ER}_{01}$ , and let  $\langle H \rangle_{\text{ER}_{01}} = \{P(\cdot|H, \beta) | \beta \in \Gamma_{\text{nat}}(H)\}$  be the restriction of  $\langle \mathcal{M} \rangle_{\text{ER}_{01}}$  to models with fixed  $H \in \mathcal{M}$ . Substituting  $\beta \equiv \ln(1 - \theta) - \ln(\theta)$  in definition 6.5, we find that  $\langle H \rangle_{\text{ER}_{01}}$ , the class of Bernoulli

models, containing one model for each probability of error

$$\begin{aligned} E_\beta(\text{ER}_{01}(Y|H, X)) &= P_\beta\{\text{ER}_{01}(Y|H, X) = 1\} = P_\beta\{H(X) \neq Y\} \\ &= \frac{1}{Z(\beta)} \exp(\beta \cdot 1) = \theta, \end{aligned} \quad (6.11)$$

and

$$\begin{aligned} E_\beta(|1 - \text{ER}_{01}(Y|H, X)|) &= P_\beta\{\text{ER}_{01}(Y|H, X) = 0\} = P_\beta\{H(X) = Y\} \\ &= \frac{1}{Z(\beta)} \exp(-\beta \cdot 0) = 1 - \theta. \end{aligned} \quad (6.12)$$

It follows that the class  $\langle \mathcal{M} \rangle_{\text{ER}_{01}}$  can equivalently be parameterized as  $\langle \mathcal{M} \rangle_{\text{ER}_{01}} = \{P(\cdot|H, \theta, \cdot) | H \in \mathcal{M}, 0 < \theta < 1\}$ , such that, if  $\text{ER}_{01}(y^n|H, x^n) = k$ , then

$$P(y^n|H, \theta, x^n) = \theta^k (1 - \theta)^{n-k}. \quad (6.13)$$

This expresses that the probability of error of  $H$  is equal to  $\theta$  for each observation, independently of any other observations. Equation (6.11) shows that, if  $(H, \beta)$  is inferred from data  $D$ , then  $\beta$  (which determines  $\theta$ ) can be interpreted as an estimate of the expected 0/1-error of  $H$ , which is just the probability that  $H$  misclassifies  $D$ . Just as above,  $\beta$  serves to estimate the expected (in this case, 0/1-) error.  $\diamond$

MDL is usually applied to concept classes in a way that does not involve entropification [83]. In example 6.5, we show that the ‘traditional’ way of applying MDL to a concept class  $\mathcal{M}$  is essentially equivalent to applying MDL to the probabilistic class  $\langle \mathcal{M} \rangle_{\text{ER}}$ , thus reconciling the two views.

### Example 6.3 (Entropification of probabilistic model classes)

What happens if we try to entropify a probabilistic model class  $\mathcal{M}$ ? For simplicity, we only consider the case where  $\mathcal{M} = \{P(\cdot|\eta) | \eta \in \Gamma_{\mathcal{M}}\}$  is a class of i.i.d. probabilistic models over  $\mathbb{E}_y$ . Similarly, we consider only error functions  $\text{ER} : \mathbb{E}_y \times \mathcal{M} \rightarrow \mathbb{R}$ . The values of  $x_i$  are therefore irrelevant and  $\mathcal{M}$  consists of full rather than conditional probability distributions. Definition 6.5 is seen to simplify in this case to

$$\begin{aligned} \langle \mathcal{M} \rangle_{\text{ER}} &= \{P(\cdot|(\eta, \beta)); \eta \in \Gamma_{\mathcal{M}}; \beta \in \Gamma_{\text{nat}}(\eta)\}, \text{ where} \\ P(y|(\eta, \beta)) &= \frac{1}{Z_\eta(\beta)} \exp(-\beta \text{ER}(y|\eta)), \\ P(y^n|(\eta, \beta)) &= \prod_{i=1}^n P(y_i|(\eta, \beta)). \end{aligned} \quad (6.14)$$

A natural error function for probabilistic models is the logarithmic error  $\text{ER}_{lg}(y^n|\eta) = -\sum_{i=1}^n \ln(P(y_i|\eta))$ . Using this logarithmic error, we obtain

$$P(y|\eta, \beta) = \frac{1}{Z_{\eta(\beta)}} \exp(\beta \ln(P(y|\eta))) = \frac{P(y|\eta)^\beta}{\sum_{y \in \mathbb{E}_y} P(y|\eta)^\beta}. \quad (6.15)$$

We consider two cases:

1.  $\mathcal{M}$  is an exponential family. A  $k$ -parameter exponential family is a family of probability distributions or densities that can be written in the form

$$P(x|\beta) = \frac{1}{Z(\beta)} \exp(-\beta^T \cdot \phi(x))h(x), \quad (6.16)$$

where  $Z(\beta) = \sum_{x \in \mathbb{E}} \exp(-\beta^T \phi(x))h(x)$ ,  $\phi(x) = (\phi_1(x), \dots, \phi_k(x))$ , and  $\beta \in \mathbb{R}^k$ .  $\phi(x)$  are functions defined for all  $x \in \mathbb{E}$ . The natural parameter space of an exponential family is given by

$$\Gamma_{nat} = \{\beta \in \mathbb{R}^k | Z(\beta) < \infty\}.$$

An exponential family is said to be full if it contains a model for every  $\beta \in \Gamma_{nat}$ . The dimension of an exponential family is the dimension of its associated  $\Gamma_{nat}$ . An exponential family is said to be of irreducible dimension if there is no  $(k-1)$ -parameter exponential family expressing the same class of probability distributions. From the point of view of measure theory, the function  $h(x)$  may be absorbed in a dominating measure [63]. One can drop the factor  $h(x)$  from (6.16). We see that maximum entropy model classes and exponential model classes coincide. If  $\mathcal{M}$  is a full exponential family then  $\langle \mathcal{M} \rangle_{ER_{I_g}} = \mathcal{M}$  as can be seen from substituting (6.16) in (6.15). If  $\mathcal{M}$  is an exponential family that is not full and that contains a model for some  $\beta \neq 0$ , then entropification serves to make it full. We see that full exponential families, and hence ‘full’ maximum entropy model classes, are closed under entropification.

2.  $\mathcal{M}$  is not an exponential family. This case is more interesting. Many useful probabilistic model classes are not of the exponential form; as a simple example consider hidden Markov models [60]. For such models classes, entropification can nevertheless be useful, for two reasons: (1) it leads to reliable estimates of the logarithmic error in the sense of definition 6.2, and (2), using  $\langle \mathcal{M} \rangle_{ER_{I_g}}$  instead of  $\mathcal{M}$  can often lead to additional compression of the data when data is encoded using the MDL two-part code. For a discussion on MDL two-part code see for example [92].  $\diamond$

We continue with presenting some useful properties of entropified model classes  $\langle \mathcal{M} \rangle_{ER}$ . These properties will be used in the proofs of our main results. The key to proving all the properties is that for each fixed  $H \in \mathcal{M}$ , the subclass of model  $\langle H \rangle_{ER}$  containing  $(H, \beta)$  for all  $\beta \in \Gamma_{nat}$  (i.e.,  $\langle H \rangle_{ER} \equiv \{(H, \beta) | (H, \beta) \in \langle \mathcal{M} \rangle_{ER}\}$ ) is essentially (though not strictly) a maximum entropy model class. The reason that the correspondence is not strict is that  $\langle H \rangle_{ER}$  is a class of conditional models. This leads to some technical, but not essential complications in proving the properties.

We assume that we are given a sample space  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$ , a model class  $\mathcal{M}$  and an error function  $ER$ . Briefly we will show that:

1. even though the distributions indexed by  $(H, \beta)$  are conditional, one can define entropy and expectation of error with respect to these distributions,
2. entropification leads to ‘reliable’ estimates of the error (in the sense of definition 6.2),
3. for fixed  $H$ , the models  $(H, \beta)$  are all, in an important sense, equivalent, but they differ in that they all have different entropy, and

4. there exists a particularly well-behaved class of error functions which we will call 'simple'.

In proving them, we need to use the maximum likelihood estimator for fixed hypotheses  $H$  and similarly, for fixed  $\beta$ , which we now define:

**Definition 6.6**

Let  $D = (x^n, y^n) \in \mathbb{E}^n$ . The maximum likelihood estimator of  $D$  for fixed  $H$  with respect to  $\langle \mathcal{M} \rangle_{\text{ER}}$ , denoted by  $\hat{\beta}(D|H)$ , is (if it exists) given by

$$\hat{\beta}(D|H) = \max_{\beta \in \Gamma_{\text{nat}}(H)} \{P(y^n|H, \beta, x^n)\}. \quad (6.17)$$

The maximum likelihood estimator of  $D$  for fixed  $\beta$  with respect to  $\langle \mathcal{M} \rangle_{\text{ER}}$ , denoted by  $\hat{H}(D|\beta)$ , is (if it exists) given by

$$\hat{H}(D|\beta) = \max_{H \in \mathcal{M}} \{P(y^n|H, \beta, x^n)\}. \quad (6.18)$$

We assume that  $Z_H(\beta)$  does not depend on  $x$ . As will be shown this implies that for fixed  $H$  and  $\beta$ , the expectation of the error under  $(H, \beta)$  is independent of the given  $x$ . The same holds for the entropy. Formally, we let  $E_{(H,\beta)}$  denote expectation under the model  $P(\cdot|H, \beta, \cdot) \in \langle \mathcal{M} \rangle_{\text{ER}}$ . Then for all  $(H, \beta) \in \langle \mathcal{M} \rangle_{\text{ER}}$  and  $x_1, x_2 \in \mathbb{E}_x$ , we have

$$E_{(H,\beta)}(\text{ER}(Y|H, X)|X = x_1) = E_{(H,\beta)}(\text{ER}(Y|H, X)|X = x_2). \quad (6.19)$$

Also, for all  $x_1, x_2 \in \mathbb{E}_x$ , the entropy  $\mathcal{H}(P(\cdot|H, \beta, \cdot))$  satisfies

$$\mathcal{H}(P(\cdot|H, \beta, x_1)) = \mathcal{H}(P(\cdot|H, \beta, x_2)). \quad (6.20)$$

They imply that the expectation  $E_{(H,\beta)}(\text{ER}(Y|H, X))$  over the conditional model  $(H, \beta)$  supplied with an arbitrary distribution  $P_x$  over  $\mathbb{E}_x$  does not depend on  $P_x$ . This allows us to write  $E_{(H,\beta)}(\text{ER}(Y|H, X))$  instead of  $E_{(H,\beta)}(\text{ER}(Y|H, X)|X = x)$ . Similarly, we will write  $\mathcal{H}(H, \beta)$  instead of  $\mathcal{H}(P(\cdot|H, \beta, x))$ .

The following proposition lists some very useful (and well-known) facts about maximum entropy/exponential model classes that will be used several times.

**Proposition 6.1**

Let  $\mathcal{M}_{\text{me}}$  be a maximum entropy class for the function  $\phi(x) = (\phi_1(x), \dots, \phi_m(x))$  with range  $\mathbb{U} = \mathbb{U}_1 \times \dots \times \mathbb{U}_m$ . Let  $\beta = (\beta_1, \dots, \beta_m) \in \Gamma_{\text{nat}}$ , where  $\Gamma_{\text{nat}}$  is the space of parameters in the natural parameterization of  $\mathcal{M}_{\text{me}}$ . Let  $1 \leq i, j \leq m$ . Then, we get

1. The first two (central) moments of  $P(\cdot|\beta)$  are determined by the first two derivatives of  $Z(\beta)$

$$\begin{aligned} \frac{\partial}{\partial \beta_i} \ln(Z(\beta)) &= -E_{\beta}(\phi_i(x)), \\ \frac{\partial^2}{\partial \beta_i \partial \beta_j} \ln(Z(\beta)) &= \text{cov}(\phi_i(X), \phi_j(X)) \\ &= E((\phi_i(X) - E(\phi_i(X)))(\phi_j(X) - E(\phi_j(X)))). \end{aligned}$$

2. Let  $\beta_1, \dots, \beta_m$  all be fixed except  $\beta_i$ . Then
  - (a)  $E_\beta(\phi(X))$  as a function of  $\beta_i$  is strictly decreasing.
  - (b) If  $\beta_i > 0$  then the entropy  $\mathcal{H}(\beta)$  is a strictly decreasing function of  $\beta_i$ . If  $\beta_i < 0$ , then  $\mathcal{H}(\beta)$  is a strictly increasing function of  $\beta_i$ .
3. The log-likelihood  $\ln(P(x^n|\beta))$  as a function of  $\beta_i$  is concave, reaching its maximum at the point where  $E_\beta(\phi_i(X)) = \overline{\phi_i(X)}^n$ . More generally
4. Let  $E_\theta$  stand for the expectation under the model  $P(\cdot|\theta) \in \mathcal{M}_{me}$  defined by the mean-value parameterization (i.e.  $E_\theta(\phi(X)) = \theta$ ). Assume that  $\overline{\phi(X)}^n$  lies in the interior of  $\mathbb{U}$ . Then

$$E_{\hat{\beta}(x^n)}(\phi(X)) = E_{\theta(\hat{x}^n)}(\phi(X)) = \overline{\phi(X)}^n = \hat{\theta}(x^n). \quad (6.21)$$

5.  $\theta(\beta) \equiv E_\beta(\phi(X))$  as a function of  $\beta$  is a continuous bijection from  $\Gamma_{nat}$  to  $\text{int}(\mathbb{U})$ .

*Proof* proposition 6.1: All of these properties are straightforward to verify by differentiation and realizing that when we take derivatives of  $Z(\beta)$  we are allowed to interchange the order of differentiation and integration by our regularity conditions on  $\phi$ . Otherwise, see [63].  $\square$

Let  $Q$  and  $R$  be two distributions over  $\mathbb{E}$  satisfying  $E(\phi(X)) = t$ . Let  $P_{me}$  be the maximum entropy distribution for this constraint. We have

$$\begin{aligned} H(Q) &\stackrel{(1)}{=} E_Q(-\ln(Q(X))) \stackrel{(2)}{\leq} E_Q(-\ln(P_{me}(X))) \\ &\stackrel{(3)}{=} E_{me}(-\ln(P_{me})) \\ &\stackrel{(4)}{=} \mathcal{H}(P_{me}) \stackrel{(5)}{\leq} E_{me}(-\ln(R(X))). \end{aligned} \quad (6.22)$$

If  $Q \neq P_{me}$ , inequality (2) becomes strict; if  $R \neq P_{me}$  inequality (5) becomes strict.

*Proof* (6.22): (1) and (4) follow from the definition of entropy, which is given by

$$\mathcal{H}(P) = E_P(L_P(x)). \quad (6.23)$$

(2) and (5) follow from the information inequality, which is given by

$$D(P||Q) \geq 0. \quad (6.24)$$

To see that  $P_{me}$  indeed maximizes the entropy subject to the constraint  $\phi(x) = t$ , let  $Q$  be any distribution other than  $P_{me}$  satisfying the constraint and notice that

$$\begin{aligned} \mathcal{H}(Q) &= E_Q(-\ln(Q(X))) < E_Q(-\ln(P_{me}(X))) \\ &= E_Q(\beta^T \phi(X) + \ln(Z(\beta))) = \beta^T E_Q(\phi(X)) + \ln(Z(\beta)) \\ &\stackrel{(6)}{=} \beta^T t + \ln(Z(\beta)), \end{aligned} \quad (6.25)$$

where the inequality follows from the information inequality (6.24) and (6) follows from the fact that we defined  $Q$  to satisfy the constraint and hence  $E_Q(\phi(X)) = t$ . On the other hand, we see

$$\begin{aligned} \mathcal{H}(P_{me}) &= E_{me}(-\ln(P(X))) = E_{me}(\beta^T \phi(X) + \ln(Z(\beta))) \\ &= \beta^T E_{me}(\phi(X)) + \ln(Z(\beta)) \\ &\stackrel{(7)}{=} \beta^T t + \ln(Z(\beta)), \end{aligned} \quad (6.26)$$

where (7) follows from the fact that, by definition, the maximum entropy distribution satisfies the constraint and hence  $E_{me}(\phi(X)) = t$ . Together, (6.25) and (6.26) give that  $\mathcal{H}(P_{me}) > \mathcal{H}(Q)$  for all  $Q \neq P_{me}$  satisfying the constraint. Now (3) follows from equation (6.25) and (6.26). (1) - (5) imply  $\mathcal{H}(Q) \leq \mathcal{H}(P_{me})$  which expresses the fact that  $P_{me}$  maximizes the entropy. It also implies that  $E_Q(-\ln(P_{me}(X))) \leq E_{me}(-\ln(R(X)))$  which expresses the fact that  $P_{me}$  minimizes the worst-case description length.  $\square$

Let  $\mathcal{M}$  be a class of models,  $ER$  be an error function and  $\langle \mathcal{M} \rangle_{ER}$  be the entropification of  $\mathcal{M}$ .  $\langle H \rangle_{ER}$ , the subclass of models from  $\mathcal{M}$  restricted to fixed  $H$  (i.e.  $\langle H \rangle_{ER} \equiv \{(H, \beta) | (H, \beta) \in \langle \mathcal{M} \rangle_{ER}\}$ ) is essentially a maximum entropy model class. However, since  $\langle H \rangle_{ER}$  is a class of conditional models, we need to use a trick: we extend  $\langle \mathcal{M} \rangle_{ER}$  to a class of distributions over  $\mathbb{E}_x \times \mathbb{E}_y$  by supplying it with the uniform distribution over  $\mathbb{E}_x$ ; this distribution exists since we assume  $\mathbb{E}_x$  to be either finite or compact. As will be shown below, the resulting model class, which we denote by  $\langle \mathcal{M} \rangle_{ER}^u$ , is a maximum entropy model class. We then use standard results about classes to prove certain properties for  $\langle \mathcal{M} \rangle_{ER}^u$ , and we then show that these properties hold for  $\langle \mathcal{M} \rangle_{ER}$ , they must also hold for the class of conditional distributions  $\langle \mathcal{M} \rangle_{ER}$ . This will be done in lemma 6.1 below. After having proved the lemma, we will show that some properties follow as immediate corollaries from this lemma.

### Lemma 6.1

Let  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$  and let  $ER : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{R} \cup \{\infty\}$  be an error function. Let  $P^u(\cdot)$  be the uniform distribution over  $\mathbb{E}_x$ . Let  $\langle \mathcal{M} \rangle_{ER}^u$  be the class of probabilistic models  $P^u(\cdot, \cdot | H, \beta)$  where for each  $(H, \beta)$  in  $\mathcal{M}$ , for each  $(x^n, y^n)$ ,  $P^u(x^n, y^n | H, \beta) = P(y^n | H, \beta, x^n) P^u(x^n)$ . We have

1. There exists a constant  $c \in \mathbb{R}$  such that for all  $n$ , all  $(x^n, y^n) \in \mathbb{E}^n$

$$P(y^n | \beta, H, x^n) \cdot c^n = P^u(x^n, y^n | \beta, H). \quad (6.27)$$

Let, for fixed  $H \in \mathcal{M}$ ,  $\langle H \rangle_{ER}^u = \{(H, \beta) | (H, \beta) \in \langle \mathcal{M} \rangle_{ER}^u\}$  be the restriction of  $\langle \mathcal{M} \rangle_{ER}^u$  to models with fixed  $H$ .

2.  $\langle H \rangle_{ER}^u$  is the maximum entropy model class for function  $\phi(x, y) \equiv ER(y | H, x)$  with range  $\mathbb{U}$ . Here  $\mathbb{U}$  is the smallest (open or closed) interval in  $\mathbb{R}$  such that  $\forall (x, y) \in \mathbb{E}: \phi(x, y) \in \mathbb{U}$ .

Let  $x$  be an arbitrary element of  $\mathbb{E}_x$ . Let  $P(\cdot | H, \beta, x)$  be the distribution over  $\mathbb{E}_y$  given by  $P(y | H, \beta, x) = Z_H(\beta)^{-1} \exp(-\beta ER(y | H, x))$  and let  $\mathcal{H}(P(\cdot | H, \beta, x))$  stand for the entropy of the distribution  $P(\cdot | H, \beta, x)$ .

3. We have for all  $(H, \beta)$

$$\begin{aligned} E_{P^u(\cdot, \cdot | H, \beta)}(ER(Y | H, X)) &= E_{P(\cdot | H, \beta, x)}(ER(Y | H, X) | X = x) \\ &= -\frac{\partial}{\partial \beta} \ln(Z_H(\beta)) \end{aligned} \quad (6.28)$$

4. We also have for all  $(H, \beta)$

$$\mathcal{H}(P(\cdot | H, \beta, x)) = \mathcal{H}(P^u(\cdot, \cdot | H, \beta)) + \ln(c). \quad (6.29)$$

*Proof lemma 6.1:* Item 1 is straightforward. Item 2 follows directly by our assumptions on ER and the definition of  $\langle \mathcal{M} \rangle_{\text{ER}}$  (definition 6.5). To prove item 3, note that for each  $P(\cdot|H, \beta, \cdot)$  the corresponding unconditional model  $P^u(\cdot, \cdot|H, \beta)$  is given by

$$P^u(x, y|H, \beta) = \frac{c}{Z_H(\beta)} \exp(-\beta \text{ER}(y|H, x)).$$

Since  $\langle \mathcal{M} \rangle_{\text{ER}}^u$  is maximum entropy model class, we have

$$E_{P^u(\cdot, \cdot|H, \beta)}(\text{ER}(Y|H, X)) = -\frac{\partial}{\partial \beta} (\ln(Z_H(\beta)) - \ln(c)) = -\frac{\partial}{\partial \beta} \ln(Z_H(\beta)). \quad (6.30)$$

Now choose an arbitrary  $x \in \mathbb{E}_x$ . Let  $\langle H(x) \rangle_{\text{ER}}$  be the class of models containing, for each  $\beta \in \Gamma_{\text{nat}}(H)$ , the distribution  $P(\cdot|H, \beta, x)$  defined as in the statement of item 3 in the lemma.  $\langle H(X) \rangle$  is a class of maximum entropy models for function  $\psi(y) \equiv \text{ER}(y|H, x)$  (note that  $\psi$  is a function of  $y$  only;  $x$  is kept fixed). This is straightforward to verify by our assumptions on ER. We now have

$$E_{P(\cdot|H, \beta, x)}(\text{ER}(Y|H, X)|X = x) = -\frac{\partial}{\partial \beta} \ln(Z_H(\beta)). \quad (6.31)$$

(6.30) and (6.31) coincide. Since we picked  $x$  arbitrarily, (6.28) follows. We continue with item 4. By straightforward calculation we see that the entropy of  $P(\cdot|H, \beta, x)$  is equal to

$$\beta E_{P(\cdot|H, \beta, x)}(\text{ER}(Y|H, X)|X = x) + \ln(Z(\beta)),$$

while the entropy of  $P(\cdot, \cdot|H, \beta)$  is given by

$$\beta E_{P(\cdot, \cdot|H, \beta)}(\text{ER}(Y|H, X) + \ln(Z(\beta)) - \ln(c)).$$

Together with (6.28) in item 3 of the lemma, equality (6.29) follows.  $\square$

We proceed to show that entropification leads to reliable estimates of the error. This will be the key to proving the theorems on entropification which we prove below. In example 6.1 we discussed why ‘reliability’ is a desirable property.

### Proposition 6.2 (reliability)

Let  $D = (x^n, y^n)$ .  $E_{(H, \beta)}(\text{ER}(Y|H, X))$  is as a function of  $\beta$ , for each  $H \in \mathcal{M}$ , continuous; moreover

$$E_{(H, \hat{\beta}(D|H))}(\text{ER}(Y|H, X)) = \frac{1}{n} \sum_{i=1}^n \text{ER}(y_i|H, x_i). \quad (6.32)$$

This proposition shows that, for each model  $(H, \hat{\beta}(D|H))$ , its expected error over future data is equal to its average error over the given data. By definition 6.2, this implies that for each  $H \in \mathcal{M}$ , the average error  $\text{ER}(y|H, x)$  can be reliably estimated on the basis of the restriction of the class  $\langle \mathcal{M} \rangle_{\text{ER}}$  to models containing this specific  $H$ .

*Proof proposition 6.2:* Let  $H \in \mathcal{M}$  be fixed. By lemma 6.1, item 1, the probabilistic model  $P^u(\cdot, \cdot|H, \hat{\beta})$  that maximizes, for fixed  $H$ , the likelihood of  $D$  within the class of unconditional models  $\langle H \rangle_{\text{ER}}^u$  (as defined in lemma 6.1) is indexed by the same value



$\hat{\beta}$  as the model  $P(\cdot|H, \hat{\beta}, \cdot)$  that maximizes the likelihood of  $D$  within the class of conditional models  $\langle H \rangle_{\text{ER}}$ . Also by lemma 6.1,  $\langle H \rangle_{\text{ER}}^u$  is a maximum entropy model class. Therefore, we have that the expectation under the unconditional model indexed by  $H$  and  $\hat{\beta}(D|H)$  is equal to the average over data  $D$

$$E_{P^u(\cdot, \cdot|H, \hat{\beta}(D|H))}(\text{ER}(Y|H, X)) = \frac{1}{n} \sum_{i=1}^n \text{ER}(y_i|H, x_i). \quad (6.33)$$

By lemma 6.1, item 3, this shows (6.32).  $\square$

We now state two properties that explain why we have chosen the name ‘entropification’. Let  $H \in \mathcal{M}$  be arbitrary but fixed. The models  $(H, \beta)$  in  $\langle \mathcal{M} \rangle$  are, for all  $\beta \in \Gamma_{\text{nat}}(H)$  except  $\beta = 0$ , partially equivalent to  $H$  as stand-alone in the sense that they leave the ordering (in terms of goodness-of-fit) that they impose on the data unchanged. The ordering with respect to the original error function equals the new ordering with respect to the logarithmic error. Yet the models  $(H, \beta)$  are all different in the sense that they all have different entropies.

### Proposition 6.3

Let  $H \in \mathcal{M}$  and let  $x^n, y^n$  and  $z^n$  be such that  $\text{ER}(y^n|H, x^n) > \text{ER}(z^n|H, x^n)$ . Then for all  $\beta \in \Gamma_{\text{nat}}(H)$  with  $\beta > 0$

$$-\ln(P(y^n|H, \beta, x^n)) > -\ln(P(z^n|H, \beta, x^n)),$$

while for all  $\beta < 0$

$$-\ln(P(y^n|H, \beta, x^n)) < -\ln(P(z^n|H, \beta, x^n)).$$

*Proof* proposition 6.3: Immediate from definition 6.5.  $\square$

Hence for each  $H$ , entropification either leaves unchanged or reverses the ordering in terms of goodness-of-fit that  $H$  imposes on the data: for every  $\beta$ , the ordering with respect to  $\text{ER}(\cdot|H, x^n)$  is identical or reversed to the ordering with respect to the code length (or ‘logarithmic error’)  $-\ln(P(\cdot|H, \beta))$ . For the second property, let  $\mathcal{H}(H, \beta)$  denote the entropy of the model  $P(\cdot|H, \beta, x)$  (for arbitrary  $x$ ) restricted to single outcomes  $\mathbb{E}_y$ .

### Proposition 6.4

For all  $\beta \in \Gamma_{\text{nat}}(H)$  with  $\beta > 0$ , the entropy  $\mathcal{H}(H, \beta)$  is a strictly decreasing function of  $\beta$ . For all  $\beta < 0$ ,  $\mathcal{H}(H, \beta)$  is a strictly increasing function of  $\beta$ .

Propositions 6.3 and 6.4 tell us that for fixed  $H$  and varying  $\beta$ , the compound models  $(H, \beta)$  can all be seen as ‘versions’ of  $H$  with different entropies.

*Proof* proposition 6.4: We know from lemma 6.1, item 4 that  $\mathcal{H}(H, \beta) = \ln(c) + \mathcal{H}(P^u(\cdot, \cdot|H, \beta))$  for some constant  $c$ . The second term stands for the entropy of a maximum entropy distribution with parameter  $\beta$  (lemma 6.1, item 2). Since for maximum entropy distributions, the entropy is a strictly increasing (decreasing) function for  $\beta > 0$  ( $\beta < 0$ ), the result follows (proposition 6.1, item 2(b)).  $\square$

Suppose  $\beta > 0^*$ . As  $\beta$  increases, the entropy  $\mathcal{H}(H, \beta)$  decreases. One may expect that, with decreasing entropy (and thus decreasing ‘inherent disorder’), the expected error  $E_{(H,\beta)}(\text{ER}(Y|H, X))$  also decreases. This relation indeed holds, but only if  $\beta > 0$ : if  $\beta < 0$ , then the entropy is an increasing function of  $\beta$  while the expected error remains a decreasing function of  $\beta$ . In general, let, for fixed  $H$ ,  $\mathbb{U}_H$  be the smallest (possibly unbounded) interval in  $\mathbb{R}$  such that  $\forall (x, y) \in \mathbb{E}: \text{ER}(y|H, x) \in \mathbb{U}_H$ .

### Proposition 6.5

$E_{(H,\beta)}(\text{ER}(Y|H, X))$  is a strictly decreasing function of  $\beta$ . For each  $t$  in the interior of  $\mathbb{U}_H$  there exists a unique value of  $\beta$  such that  $E_{(H,\beta)}(\text{ER}(Y|H, X)) = t$ .

*Proof* proposition 6.5: We know from lemma 6.1, item 3 that for all  $\beta \in \Gamma_{\text{nat}}(H)$ ,  $E_{(H,\beta)}(\text{ER}(Y|H, X)) = E_{P^u(\cdot, \cdot|H,\beta)}(\text{ER}(Y|H, X))$ . Here,  $P^u(\cdot, \cdot|H, \beta)$  (see lemma 6.1, item 2) is a maximum entropy distribution with natural parameter  $\beta$ . By applying proposition 6.1, item 2(a), which states that  $E_{P^u(\cdot, \cdot|H,\beta)}(\text{ER}(Y|H, X))$  is a strictly decreasing function of  $\beta$ , the first part follows. The second part immediately follows from item 5 of proposition 6.1.  $\square$

Some error functions, among which the squared error and the 0/1-error, turn out to have a useful additional property which automatically makes them satisfy our regularity conditions for error functions and which makes sure that entropification leaves relative ordering of hypotheses in terms of goodness-of-fit for given data unchanged. We call such hypotheses simple:

### Definition 6.7

If  $\text{ER}$  is such that for all  $H_1, H_2 \in \mathcal{M}$  and all  $\beta$ ,

$$Z_{H_1}(\beta) = Z_{H_2}(\beta),$$

where  $Z_H(\beta)$  is defined as in (6.6), then we call  $\text{ER}$  a simple error function for  $\mathcal{M}$ .

The two error functions we have encountered earlier are both simple, as shown by the following proposition.

### Proposition 6.6

Let  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$ , let  $\mathcal{M}$  be an arbitrary class of functions  $H : \mathbb{E}_x \rightarrow \mathbb{E}_y$ . If  $\mathbb{E}_y = \mathbb{R}$  then the squared error function  $\text{ER}_{sq}$  is simple for  $\mathcal{M}$ . If  $\mathbb{E}_y = \{0, 1\}$ , then the 0/1-error function  $\text{ER}_{01}$  is simple for  $\mathcal{M}$ .

*Proof* proposition 6.6: In the squared error case

$$\begin{aligned} Z_{H,x}(\beta) &= \int_{y \in \mathbb{E}_y} \exp(-\beta \text{ER}_{sq}(y|H, x)) dy \\ &= \int_{y \in \mathbb{E}_y} \exp(-\beta(y - H(x))^2) dy \\ &= \sqrt{\pi/\beta}, \end{aligned}$$

\* Note that this is always the case in statistical mechanics.

which does not depend on either  $H$  or  $x$ . The case of the 0/1-error is analogous.  $\square$

For model classes entropified with simple error functions we can drop the subscript from  $Z_H(\beta)$  and simply write  $Z(\beta)$ . By calculating the entropy  $\mathcal{H}(H, \beta)$  of the model  $(H, \beta)$  it follows immediately that this entropy depends only on  $\beta$  and not on  $H$ . Simple error functions have an additional important property which is dual to the property expressed by proposition 6.3. Whereas in that proposition, we showed that the ordering in goodness-of-fit imposed on data by a hypothesis  $(H, \beta)$  is identical for all  $\beta$  (up to their sign), the present result shows that in the case of simple error functions, a reverse property also holds: the ordering in goodness-of-fit imposed on hypothesis  $(H, \beta)$  by given data  $D$  is identical for all  $\beta$  (up to their sign).

**Proposition 6.7**

Let  $\text{ER}$  be a simple error function for  $\mathcal{M}$  and let  $D = (x^n, y^n)$ . For each  $\beta_1, \beta_2 \in \Gamma_{\text{nat}}$  with  $\beta_1, \beta_2 > 0$  or  $\beta_1, \beta_2 < 0$  and each  $H_1, H_2 \in \mathcal{M}$  we have

$$P(y^n|H_1, \beta_1, x^n) > P(y^n|H_2, \beta_1, x^n) \Rightarrow P(y^n|H_1, \beta_2, x^n) > P(y^n|H_2, \beta_2, x^n)$$

and, in particular, if  $\beta_1, \beta_2 > 0$  and if there exists a unique  $H$  that minimizes the empirical error  $\text{ER}(y^n|H, x^n)$ , then

$$\hat{H}(D|\beta_1) = \hat{H}(D|\beta_2) = \min_{H \in \mathcal{M}} \{\text{ER}(y^n|H, x^n)\}. \quad (6.34)$$

Hence the  $H$  in the tuple  $(H, \beta)$  that maximizes the likelihood is independent of  $\beta$  (except for the sign of  $\beta$ ) and (if  $\beta > 0$ ) is equal to the  $H \in \mathcal{M}$  that minimizes the empirical error.

*Proof proposition 6.7:* Immediate from instantiating  $P(y^n|H_i, \beta_i, x^n)$  using definition 6.5 and the fact that  $Z(\beta)$  does not depend on  $H$ .  $\square$

At this point, we have established all basic properties of entropification. We will use these properties to prove the main results, concerning its behaviour in the i.i.d. case. Before we do this, we give a summary of the possible interpretations of  $\beta$ . We can estimate  $\beta$  by any statistical means. The parameter  $\beta$  as part of the hypotheses  $(H, \beta)$  can be interpreted in the following ways:

1.  $\beta$  determines the expected error  $E_{(H, \beta)}(\text{ER}(Y|H, X))$ , hence ...
2. ... when  $(H, \beta)$  is inferred from data  $D$ ,  $\beta$  serves as an estimate of the error  $H$  will make on future data,
3.  $\beta$  determines the entropy  $\mathcal{H}(H, \beta)$  (proposition 6.4): the closer  $|\beta|$  to 0, the larger  $\mathcal{H}(H, \beta)$ ,
4.  $\beta$  determines how strongly the error  $\text{ER}(y^n|H, x^n)$  is weighted in the code based on  $P(\cdot|H, \beta, \cdot)$  which has lengths  $L(y^n|H, x^n) = \beta \text{ER}(y^n|H, x^n) + n \ln(Z_H(\beta))$  (equation (6.9)): the closer  $|\beta|$  to 0, the closer  $P$  is to the uniform distribution.

The last two items show that  $\beta$  can be interpreted as a kind of ‘noise’ level, measuring for each fixed  $H$  the apparent randomness of the data with respect to hypotheses  $H$ . We use the word ‘apparent’ because a small value of  $\beta$  does not mean that the data are random in any general sense; it only means that  $H$  does not give very much information about the data.

The idea to turn a function, in our case an error function  $\text{ER}(\cdot|\cdot)$ , into a class of probability distributions  $P(\cdot|\cdot, \beta) = Z(\beta)^{-1} \exp(-\beta \text{ER}(\cdot|\cdot))$  is actually not new: it is common practice in statistical mechanics [54] [86] [112] where  $P$  is the probability density function,  $Z$  the partition function, the error function is replaced by the ‘energy function’ or Hamiltonian, and instead of a parameter  $\beta$  one uses a nonnegative parameter  $T$  (called a ‘temperature’) satisfying  $\beta = 1/kT$  where  $k$  is the Boltzmann constant. Such ‘energy functions’ and ‘temperatures’ are frequently used outside a purely physical context. As far as we know, the role of the temperature  $T$  is somewhat different from our  $\beta$  since  $T$  is not treated as a parameter to be estimated from the data.

We now present our main results concerning entropification. We study the behavior of entropified model classes when data are independently distributed according to some unknown ‘true’ distribution  $P^*$ . Roughly, it is shown that with an entropified model class  $\langle \mathcal{M} \rangle_{\text{ER}}$ , if given enough data, we can find the model in  $\langle \mathcal{M} \rangle_{\text{ER}}$  with the smallest expected prediction error under  $P^*$ . Additionally, this model will provide a correct estimate of the average prediction error over future data that it will achieve; hence the model gives a good impression of ‘how good it really is’ when errors are measured by ER. The important thing is that this model which is both optimal and ‘reliable’ will be found even if  $P^*$  is not contained in  $\langle \mathcal{M} \rangle_{\text{ER}}$ . Below we state a technical lemma that is needed in our theorems. The theorems concern the general case (theorem 6.1), the case of simple error functions (theorem 6.2), the case of the logarithmic error (theorem 6.3) and the case of the squared error (theorem 6.4). The lemma and the theorems assume that the data are i.i.d. according to some unknown but fixed probability distribution  $P^*$ . We have to impose some mild conditions on  $P^*$ . These amount to the existence of some ‘window’ (i.e. a bounded set containing more than one element) within which all data will fall. The reason is that otherwise the required expectation  $E_{P^*}(\text{ER}(Y|H, X))$  may not exist. Here is a formal definition of this condition:

**Definition 6.8 (Regularity condition for the true distribution)**

Let a sample space  $\mathbb{E} = \mathbb{E}_1 \times \cdots \times \mathbb{E}_m$  be given for  $m \geq 1$ . Whenever in the following we speak of a ‘true’, or ‘generating’ distribution  $P^*$ , we assume  $P^*$  to be a distribution over  $\mathbb{E}_{P^*} = \mathbb{E}_{P^*,1} \times \cdots \times \mathbb{E}_{P^*,m}$  with full support such that for  $1 \leq i \leq m$ , (a)  $\mathbb{E}_{P^*,i} \subseteq \mathbb{E}_i$ , (b)  $\mathbb{E}_{P^*,i}$  contains more than one element and (c) if  $\mathbb{E}_i$  is continuous, then  $\mathbb{E}_{P^*,i}$  is compact.

We continue by stating our technical lemma. Essentially, it says the following: if  $\mathcal{M}$  is compactly parameterized, then the average code length of  $x^n$  based on the maximum likelihood model for  $x^n$  converges (with probability 1) to the expected code length based on the model in the class that minimizes this expected code length. This holds if the data are i.i.d. according to some  $P^*$  satisfying definition 6.8. Note that  $P^*$  is not required to be a member of  $\mathcal{M}$ .

**Lemma 6.2**

Let  $\mathcal{M} = \{P(\cdot|\theta) | \theta \in \Gamma\}$  be a class of i.i.d. probabilistic models over sample space  $\mathbb{E}$  that is finitely parameterized by  $\Gamma \subset \mathbb{R}^k$  where  $\Gamma$  is compact. Let the data be i.i.d. according to some  $P^*$  satisfying definition 6.8. Then the following minima exist for

all  $n$ , all  $x^n \in \mathbb{E}^n$ :

$$\hat{L}(x^n) \equiv \min_{\theta \in \Gamma} \{-\ln(P(x^n|\theta))\}, \quad (6.35)$$

$$\tilde{L}(P^*) \equiv \min_{\theta \in \Gamma} \{E_{P^*}(-\ln(P(X|\theta)))\}. \quad (6.36)$$

We have with  $P^*$ -probability 1:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \hat{L}(x^n) = \tilde{L}(P^*). \quad (6.37)$$

*Proof lemma 6.2:* In the proof we assume that  $\mathbb{E}$  is continuous. Adaption to the case of  $\mathbb{E}_{P^*} = \mathbb{E}_{P^*,1} \times \cdots \times \mathbb{E}_{P^*,m}$  where some of the  $\mathbb{E}_{P^*,i}$  are discrete is completely straightforward. By compactness of  $\mathbb{E}_{P^*}$  and  $\Gamma$  and the fact that we only consider  $P^*$  with associated density functions the minima (6.35) and (6.36) evidently exist. We can cover  $\Gamma$  with a grid of  $k$ -dimensional rectangles with side width  $s$ . The set  $\Gamma$  is thus partitioned into a finite number, say  $M$ , of rectangles  $R_i$ . Let, for  $1 \leq i \leq M$ ,  $\theta^i$  be the model in  $\Gamma$  corresponding to the center of  $R_i$ . In this way we obtain a reduced parameter set  $\Gamma_s = \{\theta^1, \dots, \theta^M\}$ . We first consider the simple case where  $\mathcal{M}$  is such that the following four minima are all attained by a unique value for each  $n$ ,  $x^n \in \mathbb{E}^n$ :

$$\begin{aligned} \tilde{\theta} &= \min_{\theta \in \Gamma} \{E_{P^*}(-\ln(P(X|\theta)))\}, \\ \tilde{\theta}_s &= \min_{\theta_s \in \Gamma_s} \{E_{P^*}(-\ln(P(X|\theta_s)))\}, \\ \hat{\theta}(x^n) &= \min_{\theta \in \Gamma} \{-\ln(P(x^n|\theta))\}, \\ \hat{\theta}_s(x^n) &= \min_{\theta_s \in \Gamma_s} \{-\ln(P(x^n|\theta_s))\}. \end{aligned} \quad (6.38)$$

We now show in two stages that (6.37) holds in the case where these single minimizing values exist.

*Stage 1:* Let  $n$  and  $\epsilon > 0$  be given. We claim that if we pick the rectangle side width  $s$  small enough both of the following equations will hold:

$$|E_{P^*}(-\ln(P(X|\tilde{\theta}_s))) - E_{P^*}(-\ln(P(X|\tilde{\theta}))| < \frac{1}{3}\epsilon, \quad (6.39)$$

$$|-\frac{1}{n} \ln(P(x^n|\hat{\theta}_s(x^n))) + \frac{1}{n} \ln(P(x^n|\hat{\theta}(x^n)))| < \frac{1}{3}\epsilon, \quad \forall x^n \in \mathbb{E}_{P^*}. \quad (6.40)$$

We first show (6.40). Let

$$\begin{aligned} f_n(\theta, x^n, \theta_0) &\equiv -\frac{1}{n} \ln(P(x^n|\theta)) + \frac{1}{n} \ln(P(x^n|\theta_0)) \\ &= \frac{1}{n} \sum_{i=1}^n -\ln(P(x_i|\theta)) + \ln(P(x_i|\theta_0)). \end{aligned} \quad (6.41)$$

$\mathcal{M}$  is finitely parameterized by  $\Gamma$ . Checking definition 6.3, we see that for all  $x \in \mathbb{E}_{P^*}$ ,  $f_1(\theta, x, \theta_0)$  regarded as a function of  $\theta$  must be continuous at all  $\theta \in \Gamma$ . By compactness of  $\mathbb{E}_{P^*}$  it is easy to show that

$$\begin{aligned} f_{\max}(\theta_0, \theta) &\equiv \max_{x \in \mathbb{E}_{P^*}} f_1(\theta, x, \theta_0), \quad \text{and} \\ f_{\min}(\theta_0, \theta) &\equiv \min_{x \in \mathbb{E}_{P^*}} f_1(\theta, x, \theta_0), \end{aligned}$$

are well-defined continuous functions of  $\theta$  with  $f_{\max}(\theta_0, \theta_0) = f_{\min}(\theta_0, \theta_0) = 0$ . By compactness of  $\Gamma$ , it is clear that the following function is well-defined for all  $\delta > 0$

$$g_{\max}(\delta) \equiv \max_{\mathcal{U}(\delta)} \{|f_{\max}(\theta_0, \theta)|\}, \quad (6.42)$$

where the maximum is taken over the set  $\mathcal{U}(\delta) = \{\theta, \theta_0 \in \Gamma \mid |\theta - \theta_0| \leq \delta\}$ . Moreover (compactness of  $\Gamma$ ) one can show that  $\lim_{\delta \downarrow 0} g_{\max}(\delta) = g_{\max}(0) = 0$ . The same holds for  $g_{\min}(\delta)$  which we define analogously to  $g_{\max}$ . These properties of  $g_{\max}$  and  $g_{\min}$  show the following: for every  $\epsilon > 0$ , we can pick the rectangle width  $s$  small enough such that the following implication holds for all  $\theta, \theta_0 \in \Gamma$  and all  $x^n \in \mathbb{E}_{P^*}^n$ .

$$\begin{aligned} &\text{if } \theta \text{ and } \theta_0 \text{ both fall in the same rectangle } R_i \text{ then} \\ &f_{\max}(\theta_0, \theta) < \epsilon/3 \text{ and } f_{\min}(\theta_0, \theta) > -\epsilon/3. \end{aligned} \quad (6.43)$$

It can be seen from (6.41) that for  $n \geq 1$ , for all  $x^n \in \mathbb{E}_{P^*}$

$$f_{\min}(\theta_0, \theta) \leq f_n(\theta, x^n, \theta_0) \leq f_{\max}(\theta_0, \theta). \quad (6.44)$$

(6.40) now follows by combining (6.43) and (6.44) and substituting  $\hat{\theta}(x^n)$  for  $\theta_0$ . A similar but simpler argument shows that (6.39) holds. We omit the arguments.

*Stage 2:* By the strong law of large numbers [30], we have with  $P^*$ -probability 1 that for all  $\theta_s \in \Gamma_s$ , for all  $\delta > 0$ , there exists an  $n_0$  such that if  $n \geq n_0$  then  $|\frac{1}{n} \sum \ln(P(x^n|\theta_s)) - E_{P^*}(-\ln(P(X|\theta_s)))| < \delta$ . Since  $\Gamma_s$  contains only a finite number of elements, this implies that, for all  $\epsilon > 0$ , with  $P^*$ -probability 1 there exists an  $n_0$  such that for all  $\theta_s \in \Gamma_s$

$$n \geq n_0 \Rightarrow \left| -\frac{1}{n} \sum \ln(P(x^n|\theta_s)) - E_{P^*}(-\ln(P(X|\theta_s))) \right| < \frac{1}{3}\epsilon. \quad (6.45)$$

In addition, we also have for all  $x^n \in \mathbb{E}_{P^*}$

$$E_{P^*}(-\ln(P(X|\hat{\theta}_s(x^n)))) \geq E_{P^*}(-\ln(P(X|\tilde{\theta}_s))), \quad (6.46)$$

$$-\frac{1}{n} \ln(P(x^n|\hat{\theta}_s(x^n))) \leq -\frac{1}{n} \ln(P(x^n|\tilde{\theta}_s)). \quad (6.47)$$

By first applying (6.45) with  $\theta_s \equiv \hat{\theta}_s(x^n)$  and then (6.46) we find

$$-\frac{1}{n} \ln(P(x^n|\hat{\theta}_s(x^n))) \geq E_{P^*}(-\ln(P(X|\tilde{\theta}_s))) - \frac{1}{3}\epsilon. \quad (6.48)$$

Using (6.47) and then applying (6.45) with  $\theta_s \equiv \tilde{\theta}_s$  we find

$$-\frac{1}{n} \ln(P(x^n|\hat{\theta}_s(x^n))) \leq E_{P^*}(-\ln(P(X|\tilde{\theta}_s))) + \frac{1}{3}\epsilon. \quad (6.49)$$

Combining (6.39), (6.40), (6.48) and (6.49) we find that for all  $\epsilon > 0$ , there exists an  $n_0$  such that with  $P^*$ -probability 1, for all  $n \geq n_0$

$$\left| -\frac{1}{n} \ln(P(x^n|\hat{\theta}(x^n))) - E_{P^*}(-\ln(P(X|\tilde{\theta}))) \right| < \epsilon, \quad (6.50)$$

which is equivalent to (6.37). This proves the lemma for the case that the four minima in (6.38) are all attained by single values in the parameter space.

If this is not the case we proceed as follows: by compactness all minima exist; the only problem is that they may be attained for several values. This can be handled by defining  $\tilde{\Theta}$  as the set of all  $\theta$  minimizing  $E_{P^*}(-\ln(P(X|\theta)))$  (analogously to the first line of (6.38)) and defining  $\tilde{\Theta}_s$ ,  $\hat{\Theta}$  and  $\hat{\Theta}_s$  similarly. By the same reasoning as in stage 1 of the proof, we can now prove the following existentially quantified version of (6.39) and (6.40). Let  $n$  and  $\epsilon > 0$  be given. We claim that if we pick the rectangle side width  $s$  small enough then there exist  $\tilde{\theta}_s \in \tilde{\Theta}_s$ ,  $\tilde{\theta} \in \tilde{\Theta}$ ,  $\hat{\theta}_s(x^n) \in \hat{\Theta}_s$  and  $\hat{\theta}(x^n) \in \hat{\Theta}$  such that (6.39) and (6.40) hold. By the same reasoning as in stage 2 of the proof, we can also prove a universally quantified version of (6.39) and (6.40). If  $n$  is large enough, then for all  $\hat{\theta}_s(x^n) \in \hat{\Theta}_s$  equations (6.48) and (6.49) hold with  $P^*$ -probability 1. Combining these new versions of (6.39), (6.40), (6.48) and (6.49) we can proceed as above to show that (6.50) holds. The lemma then follows.  $\square$

We now give an informal overview of the theorems we are about to prove. We start by defining an analogue of the definition of ‘reliable’ (definition 6.2) for the setting where some true (i.i.d.) distribution is assumed to exist.

#### Definition 6.9

Let the data be i.i.d. according to some distribution  $P^*$ . Let  $P$  be some given probabilistic model over  $\mathbb{E}$  and let  $\psi : \mathbb{E} \rightarrow \mathbb{U}$  be some given function. We call  $P$  reliable with respect to  $\psi$  under  $P^*$  if

$$E_P(\psi(X)) = E_{P^*}(\psi(X)).$$

A model  $P$  that is reliable with respect to  $\psi$  under  $P^*$  is (with probability 1) guaranteed to give a correct impression of the average  $\overline{\psi(x)^n}$  for large  $n$ : by the law of large numbers  $\overline{\psi(x)^n} \rightarrow E_{P^*}(\psi(X)) = E_P(\psi(X))$  as  $n$  increases, with  $P^*$ -probability 1.

Let  $\langle \mathcal{M} \rangle_{\text{ER}}$  be a model class entropified with respect to an error function ER. Let the data be i.i.d. according to some arbitrary  $P^*$  (not necessarily in  $\langle \mathcal{M} \rangle_{\text{ER}}$ ). The main point of theorem 6.1 is that for each  $H \in \mathcal{M}$ , there exists a unique  $\tilde{\beta}_H$  such that (1)  $E_{(H, \tilde{\beta}_H)}(\text{ER}(Y|H, X)) = E_{P^*}(\text{ER}(Y|H, X))$  (hence  $(H, \tilde{\beta}_H)$  is reliable with respect to ER under  $P^*$ ), and (2),  $\hat{\beta}(D|H)$ , the maximum likelihood estimator for fixed  $H$  (definition 6.6), converges with  $P^*$ -probability 1 to  $\tilde{\beta}_H$ . Hence for each  $H$ , a reliable estimate of its performance can, with probability 1 be obtained. If the error function is simple (definition 6.7), then in addition, the stronger theorem 6.2 applies. Its essence is (roughly) that the maximum likelihood estimator  $(\hat{H}, \hat{\beta})$  converges with  $P^*$ -probability 1 to the model  $(\tilde{H}, \tilde{\beta})$  where  $\tilde{H}$  is the optimal model in  $\mathcal{M}$ , minimizing the ‘true’ expected error  $E_{P^*}(\text{ER}(Y|\tilde{H}, X))$ , and  $\tilde{\beta}$  is such that  $E_{(\tilde{H}, \tilde{\beta})}(\text{ER}(Y|\tilde{H}, X)) = E_{P^*}(\text{ER}(Y|\tilde{H}, X))$ , and so  $(\tilde{H}, \tilde{\beta})$  is reliable for  $\text{ER}(\cdot|\tilde{H}, \cdot)$  under  $P^*$ . Hence the optimal  $\tilde{H}$  and a reliable estimate of its performance can both, with probability 1, be obtained. If the error function is not simple, then things get more complicated. Nevertheless, theorem 6.3 shows that in the special case of the (non-simple) logarithmic error function, an analogue to the above (maximum likelihood estimators converging to an optimal and reliable model) still holds. The squared error function is simple but satisfies an additional interesting property as will be shown in theorem 6.4.

In the theorems we make use of the maximum likelihood estimator  $\hat{\beta}(D|H)$  for fixed  $H$  which is defined in definition 6.6. It is straightforward to show that, under our conditions for the sample space of the generating distribution  $P^*$ , a unique value of  $\hat{\beta}(D|H)$  always exists. Sometimes we will also make use of the full maximum likelihood estimator  $(\hat{H}, \hat{\beta})(D)$ . In all these cases, it is straightforward to show there exists at least one maximum of the likelihood. We use the convention that, if several  $(H, \beta)$  maximums of the likelihood exist, then  $(\hat{H}, \hat{\beta})$  denotes the first one according to some prespecified ordering over  $\langle \mathcal{M} \rangle_{\text{ER}}$ .

### Theorem 6.1

Let  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$ . Let  $\mathcal{M}$  be a class of models and let  $\text{ER} : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{R}$  be an error function for  $\mathcal{M}$ . Assume that the data  $(x, y)$  are generated by independent sampling from a distribution  $P^*$  over  $\mathbb{E}_{P^*}$  as in definition 6.8. Then for all fixed  $H \in \mathcal{M}$ ,  $E_{P^*}(\text{ER}(Y|H, X))$  exists, and

1. there exists a unique  $\tilde{\beta}_H$  depending on  $H$  such that

$$E_{(H, \tilde{\beta}_H)}(\text{ER}(Y|H, X)) = E_{P^*}(\text{ER}(Y|H, X)), \quad (6.51)$$

and at the same time, for all  $\beta \in \Gamma_{\text{nat}}(H)$  with  $\beta \neq \tilde{\beta}_H$ ,

$$E_{P^*}(-\ln(P(Y|\beta, H, X))) > E_{P^*}(-\ln(P(Y|\tilde{\beta}_H, H, X))), \quad (6.52)$$

2. with  $P^*$ -probability 1

$$\lim_{n \rightarrow \infty} \hat{\beta}(x^n, y^n|H) = \tilde{\beta}_H, \quad (6.53)$$

and hence

$$\lim_{n \rightarrow \infty} E_{(H, \hat{\beta}(D|H))}(\text{ER}(Y|H, X)) = E_{P^*}(\text{ER}(Y|H, X)). \quad (6.54)$$

*Proof* theorem 6.1: We only proof the theorem for continuous  $\mathbb{E}_x$  and  $\mathbb{E}_y$ . The case where  $\mathbb{E}_x$  or  $\mathbb{E}_y$  or both are discrete is completely analogous. We first prove existence of  $E_{P^*}(\text{ER}(Y|H, X))$ . Definition 6.8 tells us that  $P^*$  is defined over a compact subspace of  $\mathbb{E}$ . Conditions C1-C3 on ER make sure that  $\text{ER}(y|H, x)$  is continuous at all  $(x, y) \in \mathbb{E}$ . For continuous  $\mathbb{E}$ , we only consider  $P^*$  with associated continuous density functions. Existence of  $E_{P^*}(\text{ER}(Y|H, X))$  now follows. Now to prove item 1, note that by definition 6.8,  $E_{P^*}(\text{ER}(Y|H, X))$  must lie in the interior of  $\mathbb{U}_H$ . Therefore there must be a unique value  $\tilde{\beta}_H$  for which (6.51) holds. We have, for each  $\beta$ ,  $E_{P^*}(-\ln(P(Y|H, \beta, X))) = \beta E_{P^*}(\text{ER}(Y|H, X)) + \ln(Z_H(\beta))$ . By differentiating with respect to  $\beta$  one verifies that  $E_{P^*}(-\ln(P(Y|H, \beta, X)))$  as a function of  $\beta$  is convex and reaches its unique minimum at the value  $\tilde{\beta}_H$  for which (6.51) holds. This proves (6.52). Concerning item 2, we will first prove (6.54). Since the data are i.i.d. we can apply the strong law of large numbers [30] which gives that with  $P^*$ -probability 1,  $n^{-1} \sum_{i=1}^n \text{ER}(y_i|H, x_i)$  converges to  $E_{P^*}(\text{ER}(Y|H, X))$ . (6.54) then follows by the reliability of estimates of ER (proposition 6.2). Relation (6.53) is now immediate by (6.54) and the fact (which we just showed) that  $E_{P^*}(-\ln(P(Y|\beta, H, X)))$  as a function of  $\beta$  is convex and reaches its single maximum at  $\beta = \tilde{\beta}_H$ .  $\square$

For simple error functions the following applies: if  $\beta > 0$  then minimization of the logarithmic error  $-\ln(P(y^n|H, \beta, x^n))$  corresponds to minimization of the error function



ER. This allows us to prove theorem 6.2, which says that the maximum likelihood estimator  $(\hat{H}, \hat{\beta})$  for data  $D$  converges to a model  $(\tilde{H}, \tilde{\beta})$  where, if  $\tilde{\beta} > 0$ , then  $\tilde{H}$  minimizes the ‘true’ expected error  $E_{P^*}(\text{ER}(Y|H, X))$  over all  $H \in \mathcal{M}$ . Let us briefly consider the case  $\tilde{\beta} < 0$ . In the case of the squared error,  $\Gamma_{\text{nat}}(H)$  only contains positive parameter values, so then always  $\tilde{\beta} > 0$  and the problem does not occur. In the special case of the 0/1-error, something interesting happens which we illustrate with an example. Suppose our concept class  $\mathcal{M}$  contains only two models  $H_1$  and  $H_2$ . Suppose  $P^*$  to be such that  $E_{P^*}(\text{ER}(Y|H_1, X)) = 0.3$  and  $E_{P^*}(\text{ER}(Y|H_2, X)) = 0.9$ . Then the hypothesis minimizing the expected 0/1-error is clearly  $H_1$ . However,  $H_2$  can be trivially modified into another ‘inverse’ hypothesis  $\bar{H}_2$  with  $E_{P^*}(\text{ER}(Y|\bar{H}_2, X)) = 0.1$ :  $\bar{H}_2(x)$  predicts 1 if  $H_2(x) = 0$  and 0 otherwise. This trivial modification can be achieved by entropification: the entropified model  $(\tilde{H}, \tilde{\beta})$  that leads to the shortest expected code length will in our example be given for  $\tilde{H} = H_2$  and  $\tilde{\beta} < 0$ ; the fact that  $\tilde{\beta} < 0$  makes  $H_2$  behave like its inverse  $\bar{H}_2$ .  $H_2$  will lead to much shorter (expected) code lengths than  $H_1$  (all this can be easily checked using (6.11) and (6.12) of example 6.2).

### Theorem 6.2

Let  $\mathbb{E}$ , data  $(x, y)$ ,  $P^*$  and  $\mathbb{E}_{P^*}$  be as in the statement of theorem 6.1. Let ER be a simple error function and assume  $\mathcal{M}$  to be such that  $\langle \mathcal{M} \rangle_{\text{ER}}$  is finitely parameterized by  $\Gamma_{\mathcal{M}} \times \Gamma_{\text{nat}}$  where  $\Gamma_{\mathcal{M}}$  is compact. Then

1. The following minima exist

$$\tilde{\text{ER}}(P^*) \equiv \min_{H \in \mathcal{M}} E_{P^*}(\text{ER}(Y|H, X)), \quad (6.55)$$

$$\tilde{L}(P^*) \equiv \min_{P(\cdot|\theta, \cdot) \in \langle \mathcal{M} \rangle_{\text{ER}}} E_{P^*}(-\ln(P(Y|\theta, X))). \quad (6.56)$$

Let  $\tilde{\theta}$  be one of the models for which the minimum in (6.56) is obtained. Then

2.  $\tilde{\theta} = (\tilde{H}, \tilde{\beta})$  for some  $\tilde{\beta} \in \Gamma_{\text{nat}}$ . If  $\tilde{\beta} > 0$  then  $\tilde{H}$  is (one of) the hypothesis (hypotheses) for which the minimum in (6.55) is obtained ( $\tilde{\beta}$  is identical for all such  $\tilde{H}$ ).

Let  $(\hat{H}, \hat{\beta}) \equiv (\hat{H}, \hat{\beta})(D)$  denote the maximum likelihood estimator in  $\langle \mathcal{M} \rangle_{\text{ER}}$ .

3. We have

$$\begin{aligned} \lim_{n \rightarrow \infty} E_{(\hat{H}, \hat{\beta})}(\text{ER}(Y|\hat{H}, X)) &= E_{(\tilde{H}, \tilde{\beta})}(\text{ER}(Y|\tilde{H}, X)) \\ &= E_{P^*}(\text{ER}(Y|\tilde{H}, X)). \end{aligned} \quad (6.57)$$

Hence, for each ‘true’, ‘generating’ distribution  $P^*$  there exists an optimal model  $(\tilde{H}, \tilde{\beta})$  such that the ‘true’ expectation under  $P^*$  of the error  $\text{ER}(Y|\tilde{H}, X)$  is minimal and equal to the expectation of this error under  $(\tilde{H}, \tilde{\beta})$ : when given enough data, every reasonable (definition 6.1) inference procedure will hit upon a model that is optimal in this sense.

*Proof theorem 6.2:* We only prove the theorem for continuous  $\mathbb{E}_x$  and  $\mathbb{E}_y$ . The case where  $\mathbb{E}_x$  or  $\mathbb{E}_y$  or both are discrete is completely analogous. Concerning items 1 and 2, existence of  $\tilde{\text{ER}}(P^*)$  is straightforward by compactness of  $\Gamma_{\mathcal{M}}$ . Existence of  $\tilde{L}(P^*)$  and item 2 will now be proven at the same time. First write

$$E_{P^*}(-\ln(P(Y|\theta, X))) = \beta E_{P^*}(\text{ER}(Y|H, X)) + \ln(Z(\beta)), \quad (6.58)$$

for  $(H, \beta) = \theta$ . Since we assume ER to be simple here,  $Z(\beta)$  does not depend on  $H$ . This shows that for each fixed  $\beta > 0$ ,  $E_{P^*}(-\ln(P(Y|(H, \beta), X)))$  reaches its minimum for the set  $\mathcal{H}^+$  of  $H$  minimizing  $E_{P^*}(\text{ER}(Y|H, X))$ . By differentiating with respect to  $\beta$  and the fact that  $Z(\beta)$  does not depend on  $H$ , one finds that there exists a single  $\tilde{\beta}^+$  minimizing  $E_{P^*}(-\ln(P(Y|(H, \beta), X)))$  for all  $H \in \mathcal{H}^+$ . For fixed  $\beta < 0$ ,  $E_{P^*}(-\ln(P(Y|(H, \beta), X)))$  reaches its minimum for the set  $\mathcal{H}^-$  of  $H$  maximizing  $E_{P^*}(\text{ER}(Y|H, X))$  (which exists by compactness of  $\Gamma_{\mathcal{M}}$ ). Now there exists a single  $\tilde{\beta}^-$  minimizing  $E_{P^*}(-\ln(P(Y|(H, \beta), X)))$  for all  $H \in \mathcal{H}^-$ . If  $\beta = 0$ , then  $E_{P^*}(-\ln(P(Y|(H, \beta), X)))$  reaches its minimum for all  $H \in \mathcal{M}$ . From this it easily follows that a  $(\tilde{\beta}, \tilde{H})$  minimizing (6.58) exists, that all minima of (6.58) have the same component  $\tilde{\beta}$  and that if  $\tilde{\beta} > 0$ , then  $\tilde{H} \in \mathcal{H}^+$ . This proves both existence of  $\tilde{L}(P^*)$  and item 2 of the theorem. The key to the proof of item 3 is the result of (6.63) below. In order to obtain this result we need to apply lemma 6.2. The lemma cannot be simply applied to model classes  $\langle \mathcal{M} \rangle_{\text{ER}}$ , since these contain conditional rather than regular probabilistic models. To avoid this problem we change  $\langle \mathcal{M} \rangle_{\text{ER}}$  into a class of essentially equivalent regular probabilistic models over  $\mathbb{E}_x \times \mathbb{E}_y$  by extending it with the uniform distribution over  $\mathbb{E}_x$  (this is possible since  $\mathbb{E}_x$  is compact by condition C4). Let  $P^u$  be the uniform distribution over  $\mathbb{E}_x$  and let

$$P^u(x^n, y^n | H, \beta) = P(y^n | H, \beta, x^n) P^u(x^n), \quad (6.59)$$

be the distribution that extends each conditional distribution  $P(\cdot | H, \beta, \cdot)$  to a full distribution over  $\mathbb{E}_x \times \mathbb{E}_y$ . We have for all  $H \in \mathcal{M}$

$$-\ln(P^u(x^n, y^n | H, \tilde{\beta})) = -\ln(P(y^n | H, \tilde{\beta}, x^n)) + C \cdot n, \quad (6.60)$$

for constant  $C$ . Here  $\tilde{\beta}$  is as in the statement of the theorem, item 2. Let

$$\tilde{L}^u(P^*) \equiv \min_{H \in \mathcal{M}} E_{P^*}(-\ln(P(y^n | H, \tilde{\beta}, x^n))) = \tilde{L}(P^*) + C, \quad (6.61)$$

where  $C$  is the same constant as in (6.60). Finally, let  $\langle \mathcal{M} \rangle_{\tilde{\beta}} = \{P(\cdot, \cdot | H, \tilde{\beta}) | H \in \mathcal{M}\}$  be the class of probabilistic models of form (6.59) for which  $\beta = \tilde{\beta}$ . It is straightforward to check that  $\langle \mathcal{M} \rangle_{\tilde{\beta}}$  is such that lemma 6.2 applies. Substituting  $\hat{L}(x^n) = -n^{-1} \ln(P^u(x^n, y^n | \hat{H}, \tilde{\beta}))$ , this gives that with  $P^*$ -probability 1

$$\lim_{n \rightarrow \infty} -\frac{1}{n} \ln(P^u(x^n, y^n | \hat{H}, \tilde{\beta})) = \tilde{L}^u(P^*). \quad (6.62)$$

Relations (6.60), (6.61) and (6.62) give us (with  $P^*$ -probability 1)

$$\lim_{n \rightarrow \infty} -\frac{1}{n} \ln(P(y^n | \hat{H}, \tilde{\beta}, x^n)) = \tilde{L}(P^*) = E_{\tilde{\beta}}(-\ln(P(Y | \tilde{H}, X))), \quad (6.63)$$

for all the  $\tilde{H}$  minimizing (6.55), where the last equality follows from item 2 in the statement of the theorem which we proved already. Exploiting the identity  $-\ln(P(y^n | H, \beta, x^n)) = \beta \text{ER}(y^n | H, x^n) + n \ln(Z(\beta))$  with  $P^*$ -probability 1, we have

$$\lim_{n \rightarrow \infty} -\frac{1}{n} \tilde{\beta} \text{ER}(y^n | \hat{H}, x^n) + \ln(Z(\tilde{\beta})) = \tilde{\beta} E_{\tilde{\beta}}(\text{ER}(Y | \tilde{H}, X)) + \ln(Z(\tilde{\beta})). \quad (6.64)$$

By reliability of estimates of ER (proposition 6.2) we have that  $n^{-1} \text{ER}(y^n | \hat{H}, x^n) = E_{(\hat{H}, \hat{\beta})}(\text{ER}(Y | \hat{H}, X))$ . Plugging this into (6.64) proves (6.57).  $\square$

It is in general difficult to analyze for non-simple error functions whether an analogue of theorem 6.2 holds. The proof of theorem 6.2 is based on the fact that, for simple error functions, minimization of logarithmic error corresponds to minimization (or maximization) of the error ER. For non-simple error functions this need not be the case since  $Z(\beta)$  varies with  $H$ . However, a special case occurs if  $\mathcal{M}$  is probabilistic and we entropify with respect to the logarithmic error. In that case, the function  $\text{ER} = \text{ER}_{lg}$  measures itself the log-likelihood of the data, while the optimal model in  $\langle \mathcal{M} \rangle_{\text{ER}}$  is also optimal with respect to expected log-likelihood. This allows an analogue to theorem 6.2 to be proven after all; it is embodied in theorem 6.3 below. In this situation, it turns out to be somewhat harder to identify exact conditions under which the required minima exist. Specifically, let  $\mathcal{M}$  be a class of probabilistic models over sample space  $\mathbb{E}$  that is finitely parameterized by  $\Gamma_{\mathcal{M}}$  and let  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  be the entropification of  $\mathcal{M}$  under the logarithmic error.  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  can be parameterized by  $\Gamma_{\mathcal{M}} \times \Gamma_{\text{nat}}(H)$ . Let data  $x_1, x_2, \dots$  be generated by independent sampling from distribution  $P^*$  over  $\mathbb{E}_{P^*}$ , where  $P^*$  is as in definition 6.8. We assume that (1)  $\mathbb{E}_{P^*}$  is such that for all  $n$ ,  $x^n \in \mathbb{E}_{P^*}^n$ , the maximum likelihood estimator of  $x^n$  in  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$ , denoted by  $\hat{\theta} \equiv \hat{\theta}(x^n)$ , exists and falls within a compact subset of  $\Gamma_{\mathcal{M}} \times \Gamma_{\text{nat}}(H)$ , and (2)

$$\tilde{L}(P^*) \equiv \min_{\theta \in \Gamma_{\mathcal{M}} \times \Gamma_{\text{nat}}(H)} E_{P^*}(-\ln(P(X|\theta))), \quad (6.65)$$

exists and is obtained by a single model  $\tilde{\theta}$ .

### Theorem 6.3

Let  $\mathcal{M}$ ,  $P^*$  and  $\tilde{\theta}$  be as above. Then it follows with  $P^*$ -probability 1

$$\begin{aligned} \lim_{n \rightarrow \infty} E_{\hat{\theta}(x^n)}(-\ln(P(X|\hat{\theta}(x^n)))) &= E_{\tilde{\theta}}(-\ln(P(X|\tilde{\theta}))) \\ &= E_{P^*}(-\ln(P(X|\tilde{\theta}))). \end{aligned} \quad (6.66)$$

*Proof* theorem 6.3: In the proof we assume the notation of example 6.3. Specifically,  $P(\cdot, \eta)$  stands for the i.i.d. probabilistic model in  $\mathcal{M}$  indexed by  $\eta$ , and  $P(\cdot | (\eta, \beta)) = Z_{\eta}^{-1}(\beta) \exp(\beta \ln(P(\cdot | \eta)))$  stands for the model in  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  indexed by  $(\eta, \beta)$ . Note also that  $\hat{\theta} = (\hat{\eta}, \hat{\beta})$ . By reliability of the estimates of  $\text{ER}_{lg}$  when the class  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  is used (proposition 6.2) we find

$$E_{(\hat{\eta}, \hat{\beta})}(-\ln(P(X|\hat{\eta}))) = -\frac{1}{n} \sum_{i=1}^n \ln(P(x_i|\hat{\eta})).$$

By straightforward calculation this gives

$$E_{(\hat{\eta}, \hat{\beta})}(-\ln(P(X|(\hat{\eta}, \hat{\beta})))) = -\frac{1}{n} \sum_{i=1}^n \ln(P(x_i|(\hat{\eta}, \hat{\beta}))). \quad (6.67)$$

Let  $\langle \mathcal{M} \rangle'_{\text{ER}_{lg}}$  be the restriction of  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  to models with parameter values in the compact set within which  $\hat{\theta}(x^n)$  must fall (we assume that such a set exist). Clearly,  $\tilde{\theta}$  must be a member of this set. Now we can apply lemma 6.2 to  $\langle \mathcal{M} \rangle'_{\text{ER}}$ . This gives, with probability 1

$$\lim_{n \rightarrow \infty} -\frac{1}{n} \ln(P(x^n|\hat{\theta}(x^n))) = \tilde{L}(P^*). \quad (6.68)$$

Together, (6.67) and (6.68) show that (6.66) holds.  $\square$

In classical statistics, the problem of curve-fitting is cast in the following terms: one assumes data to be independently generated by some unknown distribution  $P^*$  and one tries to identify the function  $H^*$  (called the ‘regression function’) that, for each  $x$ , gives the expected value (the mean) of  $Y$  given that  $X = x$  (some would prefer to say: one assumes that data are generated by some function  $H^*$  with errors distributed according to  $P^*$ ). Whatever the distribution of the errors, this can be achieved by using the squared error function in the learning phase, as will be shown in theorem 6.4 below. Such results have been known for a long time [13]. For completeness and since it is not difficult, we included theorem 6.4 nevertheless.

Let  $\mathbb{E} = \mathbb{E}_x \times \mathbb{E}_y$  where  $\mathbb{E}_x \subset \mathbb{R}$  and  $\mathbb{E}_y = \mathbb{R}$ . We assume data to be generated by independent sampling from  $P^*$  as in definition 6.8. Let  $H^*(x) = E_{P^*}(Y|X = x)$ . That is,  $H^*(x)$  gives the mean of  $Y$  for each  $x \in \mathbb{E}_x$ . We will assume that  $H^*(x)$  is continuous at all  $x \in \mathbb{E}_x$ . Let  $(\sigma^*)^2 = E_{P^*}((Y - H^*(X))^2)$ . Hence  $(\sigma^*)^2$  denotes the ‘expected true variance’ of  $Y$ . We know that, for a given class  $\mathcal{M}$  of functions  $\mathbb{E}_x \rightarrow \mathbb{E}_y$ ,  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$  consists of conditional Gaussian distributions. These distributions are obtained from the natural parameterization of  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$  by substituting  $\beta = \frac{1}{2}\sigma^2$ . Under the natural parameterization, theorems 6.1 and 6.2 are applicable. The following theorem extends these theorems for the specific case of  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$ . Briefly, the only non-trivial results that are added are the following: (1) for every  $P^*$ , the optimal model  $P(\cdot|\tilde{\sigma}^2, \tilde{H}) = P(\cdot|\tilde{\beta}, \tilde{H})$  will be such that  $\tilde{H}$  is the function in  $\mathcal{M}$  that is closest (in the mean squared error sense) to the ‘true’ function  $H^*$  and (2)  $\tilde{\sigma}^2$  can be interpreted as the mean squared error of  $\tilde{H}$ . Since, for every  $P^*$ ,  $(\hat{H}, \hat{\sigma}^2)$  will converge to  $(\tilde{H}, \tilde{\sigma}^2)$  with  $P^*$ -probability 1, this implies that in the special case with  $H^* \in \mathcal{M}$ ,  $\hat{H}$  will converge to the true  $H^*$  and  $\hat{\sigma}^2$  will converge to the true variance  $(\sigma^*)^2$  with  $P^*$ -probability 1. This holds independently of whether  $P^*(\cdot|x)$  is Gaussian or not. In the following theorem, we assume models in  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$  to be specified by  $(H, \sigma^2)$  rather than  $(H, \beta)$ .

#### Theorem 6.4

Let  $\mathbb{E}$ ,  $\mathcal{M}$ ,  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$  and  $P^*$  be as above. Then

1. For all  $H \in \mathcal{M}$ ,  $E_{P^*}(\text{ER}_{sq}(Y|H, X))$  exists, and there exists a  $\tilde{\sigma}_H^2$  depending on  $H$  such that

$$\begin{aligned} E_{(H, \tilde{\sigma}_H^2)}((Y - H(X))^2) &= E_{P^*}((Y - H(X))^2) \\ &= (\sigma^*)^2 + E_{P^*}((H^*(X) - H(X))^2). \end{aligned} \quad (6.69)$$

2. Further assume  $\mathcal{M}$  to be such that  $\langle \mathcal{M} \rangle_{\text{ER}_{sq}}$  is finitely parameterized by  $\Gamma_{\mathcal{M}} \times \Gamma_{\text{nat}}$  where  $\Gamma_{\mathcal{M}}$  is compact. Then the following minimum exists

$$\begin{aligned} \tilde{\sigma}^2 &\equiv \min_{H \in \mathcal{M}} E_{P^*}((Y - H(X))^2) \\ &= (\sigma^*)^2 + \min_{H \in \mathcal{M}} E_{P^*}((H^*(X) - H(X))^2). \end{aligned} \quad (6.70)$$

Let  $\tilde{H}$  be one of the models for which the minimum in (6.70) is obtained. We have

$$\begin{aligned} \lim_{n \rightarrow \infty} E_{(\hat{H}, \hat{\sigma}^2)}(Y - \hat{H}(X))^2 &= E_{(\tilde{H}, \tilde{\sigma}^2)}((Y - \tilde{H}(X))^2) = \tilde{\sigma}^2 \\ &= (\sigma^*)^2 + \min_{H \in \mathcal{M}} E_{P^*}((H^*(X) - H(X))^2). \end{aligned} \quad (6.71)$$

In particular, if  $H^* \in \mathcal{M}$ , then  $\hat{\sigma}^2$  converges with probability 1 to the true variance  $(\sigma^*)^2$  and  $\hat{H}$  converges to the true hypothesis  $H^*$ .

*Proof* theorem 6.4: Most of item 1 of the theorem is straightforward from theorem 6.1; the only thing that still needs to be proven is the fact that

$$E_{P^*}((Y - H(X))^2) = (\sigma^*)^2 + E_{P^*}((H^*(X) - H(X))^2). \quad (6.72)$$

Item 2 follows theorem 6.2. The only thing that is not obvious from this theorem is, once more, (6.72), and additionally

$$E_{(\tilde{H}, \tilde{\sigma}^2)}((Y - \tilde{H})^2) = \tilde{\sigma}^2. \quad (6.73)$$

It is a standard fact of regression [3] (also straightforward to verify by calculation) that for all  $\sigma^2$  and  $H$ , we have  $E_{(H, \sigma^2)}((Y - H(X))^2) = \sigma^2$ . This shows (6.73). Equation (6.72) is a variation of the well-known bias-variance decomposition [40], also straightforward to prove

$$\begin{aligned} E_{P^*}((Y - H(X))^2) - E_{P^*}((Y - H^*(X))^2) &\stackrel{(1)}{=} \\ E_{P^*(X)}(E_{P^*(Y|X)}(2YH^*(X) - 2YH(X) + H(X)^2 - H^*(X)^2|X=x)) &\stackrel{(2)}{=} \\ E_{P^*}((H(X) - H^*(X))^2). \end{aligned} \quad (6.74)$$

(1) follows from using the linearity of expectation, working out the squares and conditioning on  $X$ . (2) is obtained by using  $E_{P^*}(Y|X=x) = H^*(x)$ . From (6.74), the equality (6.72) is immediate.  $\square$

## 6.2 Entropification and MDL

Is entropification merely a convenient tool to make predictions reliable or are there additional reasons as to why we should ‘entropify’ our model classes? In this section we show that if we use the MDL principle as our statistical inference procedure, then it is often a good idea to use an entropified model class for at least two different reasons: first, entropification can serve to optimize the trade-off between hypothesis complexity and goodness of fit as needed in the two-part MDL code and the stochastic complexity. Second, it leads to codes for non-probabilistic model classes that can be justified in terms of minimizing expected code lengths.

There have been different proposals in the literature on how to deal with two-part codes and stochastic complexity codes for non-probabilistic model classes in the MDL framework. For simplicity we will restrict our discussion to the two-part codes. Recall, that in the basic, probabilistic case, we select  $H$  minimizing

$$-\log(P(x^n|H)) + L_{C_1}(H), \quad (6.75)$$

where  $C_1$  is some code used for encoding the parameters indexing the hypothesis. In [92] it is proposed to turn a non-probabilistic model class into a class  $\mathcal{M}_{pr}$  (which essentially corresponds to entropification with  $\beta = 1$ ). This leads to finding the  $H$  minimizing

$$\beta \text{ER}(y^n|H, x^n) + n \ln(Z_H(\beta)) + L_{C_1}(H), \quad (6.76)$$

with  $\beta = 1$ . The problem here is that choosing  $\beta = 1$  is essentially arbitrary but can have large consequences: choosing a different value of  $\beta$  we may end up, at least for small  $n$ , with  $H$  of different complexity (the closer  $\beta$  to 0, the larger the relative weight of the complexity term on (6.76)).

In [7] it is proposed to select the hypotheses  $H$  minimizing the sum of the empirical error  $\text{ER}(y^n|H, x^n)$  and the square root of the complexity term times the sample size,  $\sqrt{nL_{C_1}(H)}$ . While this criterion can be shown to have some strong asymptotic properties, it is in a sense not faithful to the MDL principle since the resulting sum does not have a natural interpretation as a code length.

In [125] it is proposed to minimize  $\beta \text{ER}(y^n|H, x^n) + L_{C_1}(H)$  for some  $\beta$  whose value is made dependent on the size of the training set. Once more, this has strong asymptotic properties, but again, it is not clear how to interpret the resulting sum from a purely coding-theoretic point of view.

Instead, we propose to entropify  $\mathcal{M}$  and then use (6.76) (now with an additional term  $L_{C_1}(\beta)$  added to account for the number of bits needed to encode  $\beta$ ). We think there are several advantages to using entropified model classes. We first note that, at least non-asymptotically, using the entropified class  $\langle \mathcal{M} \rangle_{\text{ER}}$  can lead on to choose different models for the same data than when using  $\mathcal{M}_{pr}$  for fixed  $\beta$ . We give an example.

#### Example 6.4

Consider a class of continuous functions  $\mathcal{M}$  entropified with the squared error. Let data  $D = (x^n, y^n)$  and model  $H \in \mathcal{M}$  be given. Denote the average squared error  $H$  makes on  $D$  by  $\overline{\text{ER}}_{sq}$ . Using the code (6.76) for fixed  $\beta$ , we obtain as total description length of the  $y^n$  given the  $x^n$

$$L(y^n; H|x^n) = n \left( \overline{\text{ER}}_{sq} + \frac{1}{2} \ln \left( \frac{\pi}{\beta} \right) \right) + L_{C_1}(H), \quad (6.77)$$

while using (6.76) for the entropified model  $(H, \beta)$  where  $\beta = \hat{\beta}(D|H)$  is the parameter that maximizes the likelihood of  $D$  given  $H$ , we obtain

$$L'(y^n; H|x^n) = \frac{1}{2}n (1 + \ln(2\pi) + \ln(\overline{\text{ER}}_{sq})) + L_{C_1}(\beta) + L_{C_1}(H), \quad (6.78)$$

which depends logarithmically rather than linearly on the average error (both equations can be verified by substituting  $\beta = \frac{1}{2}\sigma^2$ ). When two-part code MDL is used, the MDL optimal  $\beta_{\text{mdl}}$  for given  $D$  and fixed  $H$  will not be equal to  $\hat{\beta}$  but nevertheless it will be reasonably close.  $L_{C_1}(\beta)$  will be equal to  $\frac{1}{2} \log(n) + c$  for some constant  $c$ . This implies that there can be very well be hypotheses  $H_1$  and  $H_2$  with different number of parameters (so  $L_{C_1}(H_1) \neq L_{C_1}(H_2)$ ) such that  $H_1$  minimizes (6.77) while  $H_2$  minimizes (6.78). In such a case, two-part code MDL based on the entropified model class leads to a different optimal  $H$ .  $\diamond$

Using (6.76) with entropified model classes  $\mathcal{M}_{\text{ER}}$  allows the sum (6.76) (now with additional term  $L_{C_1}(\beta)$ ) to be interpreted as a code length. By learning the optimal value of  $\beta$  from the data (which is what entropification in the two-part code setting amounts to), we essentially choose the value that allows the shortest code length of the data, which is in line with the general MDL philosophy. Moreover, since each model

$\langle \mathcal{M} \rangle_{\text{ER}}$  corresponds to a code, we can also define stochastic complexity with respect to such model classes in the usual way and use it as a basis of model class selection; it allows us to compare different model classes based on different error functions for the same data, since the performance of all the classes are measured using the same criterion, namely, the code length. This is once again in line with the general MDL philosophy of using code lengths as a ‘universal yardstick’ [92], to be employed whenever different models or model classes are to be compared for the same data.

Another thing to be said for entropification is that it unifies different instantiations of MDL. In the existing literature on MDL, the question of how to code the data given an hypothesis has been given different answers depending on the category of model class used. For probabilistic model classes, generally the Shannon-Fano code with  $L(D) = -\log(P(D))$  is used [8], [92]. For concept classes (classes consisting of functions  $\mathbb{E}_x \rightarrow \{0, 1\}$ ), the usual approach [83] has been to explicitly code the mistakes a hypothesis  $H$  makes on data  $D$ . For the case of non-probabilistic model classes with arbitrary error functions ER, there have been several proposals, as we saw above. Entropification (where data  $D = (x^n, y^n)$  given hypotheses  $(H, \beta)$  is encoded using the code lengths  $-\log(P(y^n|H, \beta, x^n))$ ) is an approach to handle non-probabilistic model classes that contains the existing treatments of probabilistic and concept classes as special cases. In the probabilistic case, as long as the model class is a full exponential family, then entropification will not change anything. In the case where  $\mathcal{M}$  is a concept class, the code based on entropification with respect to the 0/1-error, while superficially different, is essentially equivalent to the traditional approach of coding, i.e., the mistakes  $H$  makes on  $D$ . This suggests (but does not prove of course) that entropification can serve as the general ‘preprocessing’ tool to make a single version of two-part code MDL applicable to essentially arbitrary model classes. We now show this formally in the example below.

#### Example 6.5 (Concept learning and Bernoulli parameters)

Let  $\mathcal{M}$  be a class of concepts over  $\mathbb{E} = \mathbb{E}_x \times \{0, 1\}$  and let the observational data  $D = (x^n, y^n)$ . Two-part codes for concept classes are traditionally [83] based on the following coding scheme: the  $x_i$  are regarded as given. The  $y_i$  are encoded by first encoding an hypotheses  $H \in \mathcal{M}$  and then encoding the exceptions to  $H$ , which are all the indices  $i$  for which  $y_i \neq H(x_i)$ . We assume that hypotheses are encoded using some fixed code  $C_1 : \mathcal{M} \rightarrow B^*$ . Clearly, given the  $x_i$ ,  $H$  and the list of expectations  $M = \{i_1, \dots, i_k\}$  we can fully reconstruct  $y_1, \dots, y_n$  (for  $x_i$  with  $i \notin M$  we set  $y_i = H(x_i)$ ; for  $x_i$  with  $i \in M$  we set  $y_i = |1 - H(x_i)|$ ). If  $H$  makes  $k$  mistakes on a sample  $D$  of length  $n$ , there are  $\binom{n}{k}$  different exception sets  $M = \{i_1, \dots, i_k\}$ . Hence we need  $\ln \binom{n}{k} + L(k)$  nats to encode all these mistakes. Here  $L(k) = \mathcal{O}(\ln(k))$  equals the number of nats needed to encode  $k$  using some prefix code for the numbers  $0, \dots, n$  (note that  $k$  has to be encoded to allow unique decoding). The total description length of the  $y_i$  given the  $x_i$  becomes

$$L(y^n, H|x^n) = \ln \binom{n}{k} + L(k) + L_{C_1}(H). \quad (6.79)$$

Another way to arrive at a two-part code for the data would be to first entropify the concept class  $\mathcal{M}$  with respect to the 0/1-error function and then to encode data by first coding some  $H$ , than some parameter  $\beta$  (using a fixed code  $C'_1$  and then encoding the

$y_i$  using the code based on  $P(\cdot|H, \beta)$ . This would take

$$L(y^n, H, \beta|x^n) = -\ln(P(y^n|H, \beta, x^n) + L_{C'_1}(\beta) + L_{C'_1}(H) \quad \text{nats.} \quad (6.80)$$

We proceed to show that (6.79) and (6.80) approximately coincide and hence that both ways of coding the data are essentially equivalent. By substituting  $\beta \equiv \ln(1 - \theta) - \ln(\theta)$ , we find that the class  $\langle \mathcal{M} \rangle_{\text{ER}_{01}}$  can be parameterized as follows:  $\langle \mathcal{M} \rangle_{\text{ER}_{01}} = \{P(\cdot|H, \theta, \cdot)|H \in \mathcal{M}, 0 < \theta < 1\}$ , such that, if  $\text{ER}_{01}(y^n|H, x^n) = k$ , then

$$P(y^n|H, \theta, x^n) = \theta^k(1 - \theta)^{n-k}. \quad (6.81)$$

Instead of coding  $\beta$  we can also code the  $\theta$  corresponding to it. We can therefore rewrite (6.80) as follows

$$L(y^n, H, \theta|x^n) = -\ln(P(y^n|H, \theta, x^n)) + L_{C'_1}(\theta) + L_{C'_1}(H) \quad \text{nats,} \quad (6.82)$$

where  $P(y^n|H, \theta, x^n)$  is given by (6.81). The maximum likelihood estimator  $\hat{\theta}$  maximizing (6.81) for fixed  $H$  and  $(x^n, y^n)$  is given by  $\hat{\theta} = k/n$ . Based on  $H$  and  $\hat{\theta}$ , the number of nats  $-\ln(P(y^n|H, \hat{\theta}, x^n))$  needed to code the data becomes

$$-\ln(P(y^n|H, \hat{\theta}, x^n)) = -\ln(\hat{\theta}^k(1 - \hat{\theta})^{n-k}) \stackrel{(1)}{=} n\mathcal{H}(\hat{\theta}) \stackrel{(2)}{\approx} \ln \binom{n}{k}, \quad (6.83)$$

where (1) follows by straightforward calculation and (2) by Stirling's approximation  $\ln(n!) = n \ln(n) - n + \ln(\sqrt{2\pi n}) + \mathcal{O}(\frac{1}{n})$ . For precise bounds on  $|n\mathcal{H}(\hat{\theta}) - \ln \binom{n}{k}|$  see [20]. Since  $\hat{\theta} = k/n$  and hence, when  $n$  is known (as we assume in this example), can be reconstructed from  $k$  only, we need approximately  $L(k)$  nats to describe  $\hat{\theta}$ , where  $L(k)$  is defined as above. The total description length of the  $y_i$  then becomes

$$-\ln(P(y^n|H, \hat{\theta}, x^n)) \approx \ln \binom{n}{k} + L(k) + L_{C'_1}(H), \quad (6.84)$$

which is seen to coincide with (6.79). Hence if we code the data based on the entropified model class  $\langle \mathcal{M} \rangle_{\text{ER}_{01}}$  and use the optimal  $\beta$  (corresponding to the optimal  $\theta$ ) for given  $D$  and  $H$ , then the number of bits we need coincides with the number of bits needed to efficiently encode the exceptions.  $\diamond$

The arguments given above suggest that entropification can serve as a general means to apply two-part code MDL to non-probabilistic model classes. Of course, they do not prove that entropification will be as well-behaved as either approach described in [7] or [125] to this problem.

When a probabilistic model class is used in two-part code MDL, data is encoded by first encoding some model  $\theta$  and then coding the data based on the Shannon-Fano code  $L(x^n|\theta) = -\log(P(x^n|\theta))$ . Why to use this code and not any other one? There are many other possibilities; to give an example, we could map each model  $\theta$  to the code with lengths  $L'(x^n|\theta) = -\log(\sqrt{P(x^n|\theta)}/\sum_{z^n \in \mathbb{E}^n} \sqrt{P(z^n|\theta)})$  which, by the Kraft inequality,

$$\sum_{x \in \mathbb{E}} 2^{-L(x)} \leq 1, \quad (6.85)$$



also corresponds to a probability distribution over  $\mathbb{E}^n$ . Why does the Shannon-Fano code have a special status? The justification lies in the fact that, by using the Shannon-Fano code, the code length of the data precisely reflects the probability: if  $P(D_1|\theta) = a \cdot P(D_2|\theta)$ , then  $L(D_1|\theta) = L(D_2|\theta) + \log(a)$ .

Some authors prefer a different (or at any rate, additional) justification based on the information inequality (6.24): if  $\theta$  turns out to be the 'true' model, i.e. data is generated by repeated sampling from  $\theta$ , then the expected code length  $E_\theta(L(X^n))$  is minimized. We set  $L(x^n) \equiv -\log(P(x^n|\theta))$ . By using the Shannon-Fano code, we map each model  $\theta$  to the code that will be optimal if  $\theta$  is actually true; hence it is the code that best 'suits'  $\theta$ . This justification of the use of the Shannon-Fano code can be found in [123]. We have always had some doubts about this argument, for two reasons: (a) it does not say anything about the (realistic) case where the model class contains models that allow us to compress the data (hence we can learn something about the data); yet none of these models are close to the 'true' one generating the data; (b) it is not clear how to extend this argument to non-probabilistic model classes.

Proposition 6.8 below shows how entropification allows us to extend the Shannon-Fano argument to a more general case which includes non-probabilistic model classes. Whereas we still assume the existence of some true probability distribution generating the data, we do not assume any more that it is contained in the model class  $\mathcal{M}$  under consideration. For simplicity, we will consider only the unconditional, 'unsupervised' case, where we are interested in coding complete outcomes (and not just  $y$ -values conditioned on  $x$ -values). Formally, we consider a class  $\mathcal{M}$  of models and an error function  $\text{ER} : \mathbb{E} \times \mathcal{M} \rightarrow \mathbb{R}$  (in case  $\mathcal{M}$  is probabilistic we take  $\text{ER}$  to be the logarithmic error). We let  $\theta = (H, \beta)$  index a model in  $\langle \mathcal{M} \rangle$ . Let  $\mathcal{G}$  be the class of probability distributions  $P^*$  over  $\mathbb{E}$  satisfying

$$E_{(H,\beta)}(\text{ER}(X|H)) = E_{P^*}(\text{ER}(X|H)). \quad (6.86)$$

Let  $\mathcal{L}$  be the class of all code length functions  $L : \mathbb{E} \rightarrow \mathbb{R} \cup \{\infty\}$  satisfying the Kraft inequality (6.85). We are now ready to state our proposition. We discuss its implications after the proof.

### Proposition 6.8

Let  $\mathcal{M}$ ,  $\text{ER}$ ,  $\mathcal{G}$ ,  $\theta$  and  $\mathcal{L}$  be as above. Let  $L(\cdot|\theta) \in \mathcal{L}$  be the code length function of the Shannon-Fano code for  $\theta = (H, \beta)$ , restricted to one outcome  $x \in \mathbb{E}$ . That is

$$L(x|H, \beta) = -\log(P(x|H, \beta)) = \beta \text{ER}(x|H) + \ln(Z_H(\beta)).$$

We have:

1.

$$L(\cdot|\theta) = \inf_{L \in \mathcal{L}} \sup_{P^* \in \mathcal{G}} E_{P^*}(L(X|\theta)). \quad (6.87)$$

That is,  $L(\cdot|\theta)$  gives the shortest worst-case code lengths, the worst-case being taken over all distributions satisfying (6.86).

2. Let, for given  $H$ ,  $\mathbb{U}_H$  be the smallest interval such that  $\forall x : \text{ER}(x|H) \in \mathbb{U}_H$ . For every  $H \in \mathcal{M}$  and for every  $P^* \in \mathcal{G}$  for which  $E_{P^*}(\text{ER}(X|H))$  lies in the interior of  $\mathbb{U}_H$ , there exists a  $\beta$  such that (6.86) holds.

*Proof* proposition 6.8: Define  $t \equiv E_{(H,\beta)}(\text{ER}(X|H)) = E_{P^*}(\text{ER}(X|H))$ . As is clear from the regularity conditions for error functions, the probability distribution  $P(\cdot|H, \beta)$  is the maximum entropy distribution for the constraint  $E(\text{ER}(X|H)) = t$ . Using (6.22), we have that  $E(\text{ER}(X|H)) = E_{(H,\beta)}(L(X|H, \beta)) = \mathcal{H}(H, \beta)$  for every  $P^* \in \mathcal{G}$ . On the other hand, let  $L' \in \mathcal{L}$  be a code length function different from  $L(\cdot|H, \beta)$ . By (6.22), there exists a  $P^* \in \mathcal{G}$  (namely,  $P^* = P(\cdot|H, \beta)$ ) such that  $E_{P^*}(L'(X)) > \mathcal{H}(H, \beta)$ . This proves (1). To prove (2), note that the class of maximum entropy models for function  $\text{ER}(X|H)$  coincides with the class of models in  $\langle \mathcal{M} \rangle_{\text{ER}}$  restricted to fixed  $H$ . Let us denote this subclass by  $\mathcal{M}_{me}$ . By proposition 6.1, for each  $t$  in the interior of  $\mathbb{U}$ ,  $\mathcal{M}_{me}$  contains a model satisfying  $E(\text{ER}(X|H)) = t$ . This proves (2).  $\square$

This proposition shows that the Shannon-Fano code for models  $\theta$  in entropified model classes (a) leads to codes that are worst-case optimal if the probability distribution  $\theta$  is ‘true’ only in the sense that its expectation of the error coincides with the true expectation of error, and (b) that entropified model classes always (except possibly for  $P^*$  with expected errors at the boundaries of the error space) contain a model that is ‘true’ in this weak respect.

If one uses a non-probabilistic model class  $\mathcal{M}$ , one usually does not have a clear idea about the distribution generating the data. If one is at all willing to assume that such a distribution nevertheless exists, then it seems reasonable to make as few assumptions as possible about it. This directly leads to our worst-case scenario, which really says that every i.i.d. distribution is a possible candidate for generating the data. That is why we regard this proposition as justifying the use of the Shannon-Fano code for the entropified (probabilistic version) of  $\mathcal{M}$ . We hasten to add though that there do exist codes (based on non-i.i.d. model classes) whose expected lengths under every  $P^*$  are arbitrarily close to that of the Shannon-Fano code. An example is the code based on the universal computer language.

The proposition also has something to say about the case where  $\mathcal{M}$  itself is probabilistic and we entropify with respect to the logarithmic error  $\text{ER}_{lg}$ . If  $\mathcal{M}$  is itself an exponential family, this will not change  $\mathcal{M}$  (example 6.3), and the proposition tells us that the Shannon-Fano code for  $\mathcal{M}$  is optimal not only in the case that the data is generated by one of the models in  $\mathcal{M}$ , but also in the case it is generated by some i.i.d. model not in  $\mathcal{M}$ . If  $\mathcal{M}$  is not an exponential family, then the usual optimality of the Shannon-Fano code holds for the models  $\mathcal{M}$ , while ‘worst-case’ optimality holds, by proposition 6.8, for the models in  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$ . Whether one should entropify or not then depends on whether one thinks that one of the models in the class will be very close to being ‘truly a true model’: if one entropifies, one adds an extra dimension to the parameter space. This can lead to logarithmically larger code lengths; if  $\mathcal{M}$  contains the true model, then it will lead even with probability 1 to larger code lengths of the data, when data is encoded using either the stochastic complexity or the two-part MDL code. However, if the true model is not in  $\mathcal{M}$ , then using  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  instead of  $\mathcal{M}$  can sometimes lead to a linear decrease in code lengths. We briefly show why.

To see that if  $\mathcal{M}$  contains the true model, one will need more bits to encode the data based on  $\langle \mathcal{M} \rangle_{\text{ER}_{lg}}$  rather than  $\mathcal{M}$ . Now, we use a result described in [17]. In the paper it is proved that an analogue of the asymptotic expansion of stochastic complexity exists for the case where data is distributed according to one of the models in a (probabilistic)

model class  $\mathcal{M}$ . Let  $\mathcal{M}$  be a probabilistic model class consisting of i.i.d. models. In [17] it is shown that, if the data are generated by one of the models  $\theta^*$  in  $\mathcal{M}$ , then under some mild further conditions on  $\mathcal{M}$

$$L_{sc}(x^n|\mathcal{M}) = -\log(P(x^n|\theta^*)) + \frac{k}{2}\log(n) + \mathcal{O}(1), \quad (6.88)$$

with  $\theta^*$ -probability 1. Here  $L_{sc}(x^n|\mathcal{M})$  is the stochastic complexity of  $x^n$  with respect to  $\mathcal{M}$  and  $k$  is the number of parameters needed for parameterizing  $\mathcal{M}$ . The two-part code length is within  $\mathcal{O}(1)$  of the stochastic complexity. Observe that if the true model  $\theta^*$  is in  $\mathcal{M}$ , then it is also in  $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ . Therefore, by (6.88), using  $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ , the number of parameters  $k$  is increased by 1 which results in a logarithmic increase in code length (with probability 1). If  $\theta^*$  is not in  $\mathcal{M}$  then, supposing  $\mathcal{M}$  is finitely parameterized by  $\Gamma \in \mathbb{R}^k$ , the asymptotic expansion of both stochastic complexity and two-part code gives

$$L_{sc}(x^n|\mathcal{M}) = -\log(P(x^n|\hat{\theta}(x^n))) + \frac{k}{2}\log(n) + \mathcal{O}(1).$$

By applying lemma 6.2, one sees that with  $\theta^*$ -probability 1

$$\lim_{n \rightarrow \infty} -\frac{1}{n}L_{sc}(x^n|\mathcal{M}) = \min_{\theta \in \Gamma} E_{\theta^*}(-\ln(P(X|\theta))).$$

This will hold for both the original model class  $\mathcal{M}$  and its entropified version  $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$  (in the latter case, the values in  $\beta$  have to be restricted to a compact set). By proposition 6.8, there exists a  $\theta^*$  such that

$$\min_{\theta \in \langle\Gamma\rangle_{\text{ER}_{lg}}} E_{\theta^*}(-\ln(P(X|\theta))) < \min_{\theta \in \Gamma} E_{\theta^*}(-\ln(P(X|\theta))),$$

where we used  $\langle\Gamma\rangle_{\text{ER}_{lg}}$  to denote the parameterization of  $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ . In such a case, both the two-part and the stochastic complexity code based on  $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$  will clearly achieve more compression (by a linear amount) than the codes based on  $\mathcal{M}$ . Since  $\mathcal{M} \subset \langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ , the opposite event (the code based on  $\mathcal{M}$  achieving a linear gain in compression compared to the code based on  $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ ) has zero probability for any  $\theta^*$ .

### 6.3 Discussion

We have introduced the concept of ‘entropification’ and shown how it can be used in the context of estimating prediction error and in the context of MDL. We leave detailed conclusions to the epilogue, where we discuss how the results obtained in this chapter can be used to partially resolve the problematic issues concerning MDL.



## 7. Epilogue: using models in a careful way

In this discussion we study whether we have resolved the problematic issues concerning MDL. Briefly these are

1. Can we use models that are partially wrong to give reasonable predictions of future data? If so, what can we do with them and what not?
2. What to do if we are asked to use our model to predict future data while utilizing various loss functions?
3. When using a probabilistic model class, why should we use the Shannon-Fano code  $L(D|\theta) = -\log(P(D|\theta))$  to encode the data with the help of a model? When using a non-probabilistic model class, why should we use the code for which  $L(D|H) = \text{ER}(D|H) + K$  for all  $D$ ?

From the point of view of the MDL philosophy, we choose a model class  $\mathcal{M}$  because we think it will help us to capture some of the regularity inherent in the data - but we have no hope that it will capture all (except if  $\mathcal{M}$  corresponds to the class of all computer programs written in some language - but then the inference process becomes non-computable). This is the situation we will usually be in: all our models will always, to some extent, be wrong. Therefore, though the question came up in connection with the MDL philosophy, it should be relevant not only to MDL, but to all statistical inference procedures.

Once we accept the fact that our model  $H$  for data  $D$  will always be partially wrong, we are faced with the question of what can be reliably inferred from such a model and what not. We can always change our model classes in such a way that we can reliably estimate the prediction error over future data. This will lead, with high probability, to accurate estimates of error over future data even if the data are independently drawn according to a distribution that is completely different from our model. Hence, if we are willing to make the i.i.d. assumption, then as long as we measure the error our model makes when predicting future data using the same error function as the one that was used in inferring the model from the data, we can use the model  $H$  reliably: even though it is partially wrong, it will give a correct impression of how accurate it is in predicting future data. Note however, that we used the i.i.d. assumption - so we still have to assume something about 'the truth out there'. Hence, we have not resolved in general whether using an overly simply (that is, partially wrong) model can lead to 'disastrous' results. Prediction errors may be accurately estimated under a much wider assumption than the classical assumption that one's model class contains the true model. The question remains of whether this is not too weak; whether we will not always be interested in estimating more aspects about the data than just their prediction error. The example below shows that sometimes 'reliable' predictions are enough, while at the same time 'unreliable' predictions can lead to very misleading results.

### Example 7.1 (Classification)

Recall that in concept learning the model class consists of functions  $H : \mathbb{E}_x \rightarrow \{0, 1\}$ . Frequently the goal will be to use the concept  $\tilde{H}$  learned on the basis of data  $D$  to

classify new data: one is given a value  $x \in \mathbb{E}_x$  and one has to predict the corresponding  $y \in \{0, 1\}$ . Suppose one uses a class of concepts  $\mathcal{M}$  entropified with the 0/1-error. Suppose further that, for given data  $D$ , the estimate inferred is  $(\tilde{H}, \tilde{\theta})$ . This estimate says that if the model  $\tilde{H}$  is used, then the probability of making a wrong prediction is  $\tilde{\theta}$ . Let us assume that the data are i.i.d. according to a model  $P^*$  so that if  $D$  is large enough and one uses a reasonable estimation procedure, then  $\tilde{\theta} \approx P^*(\tilde{H}(X) \neq Y)$ . This means that if we are only interested in classifying future data, the component  $\tilde{\theta}$  of our model  $(\tilde{H}, \tilde{\theta})$  will give us a good idea on just how well we can do that. Hence if we use our model only for classification, we have neither an overly optimistic nor an overly pessimistic of how good our model is at this task. Now let us consider a specific example where the model  $(\tilde{H}, \tilde{\theta})$  as issued by our estimation procedure on the basis of a large data set  $D = (x^n, y^n)$  has  $\tilde{\theta} = 0.95$ . This means that, if  $D$  is really large, we can predict future data with 95% accuracy. However, it may be the case that, for the  $x_i$  where  $\tilde{H}(x_i) = 1$ ,  $y_i$  is always equal to 1 while for the cases where  $\tilde{H}(x_i) = 0$ ,  $y_i \neq \tilde{H}(x_i)$  half the time. If  $\tilde{H}(x_i) = 1$  in 90% of the cases, then we will have  $\tilde{\theta} \approx 0.95$  while, if a new value  $x$  is given such that  $\tilde{H} = 0$  and we use  $\tilde{H}(x)$  for prediction, we will only be right in about 50% of the cases. Hence our model is very bad for new data with  $\tilde{H}(x) = 0$ , and if a loss function is used such that predicting a ‘false zero’ leads to a much higher loss than when predicting a ‘false one’, then our model will really be quite worthless. Prediction utilizing such a loss function is not reliable - and indeed, if we stick to ‘safe statistics’, we are not allowed to make such a prediction. We used an extreme example, but similar examples, where there exists a very simple rule that gives accurate predictions for a large subset of the  $x_i$  while being quite bad on the remaining  $x_i$  do occur in practice.  $\diamond$

If we use an entropified model class  $\langle \mathcal{M} \rangle_{\text{ER}}$ , and we want to use our estimate  $(\tilde{H}, \tilde{\beta})$  to make predictions or decisions using loss functions that cannot be written as a linear combination of ER we see that, at least if the sample space  $\mathbb{E}$  is discrete, this will often still work - but it will also often be unreliable. That is because the model class  $\langle \mathcal{M} \rangle_{\text{ER}}$  restricted to models with fixed  $H$  is essentially a maximum entropy model class. In this case, it tells us that for an exponentially large majority of those data sets to which  $(\tilde{H}, \tilde{\beta})$  gives a good fit, the frequencies  $(\gamma_1, \dots, \gamma_k)$  will be approximately equal to the probabilities  $P(1|\tilde{H}, \tilde{\beta}), \dots, P(k|\tilde{H}, \tilde{\beta})$ . If future data indeed belongs to this majority, then the average of every function (hence also every loss function) over future data will be approximately equal to its expectation over  $(\tilde{H}, \tilde{\beta})$ , and the predictions will be accurate. Only in a few cases, where the frequencies and the probabilities do not coincide, the predictions will not be accurate - nevertheless, as we saw in the example above, these cases may certainly occur.

This leaves us with justifying the use of the Shannon-Fano code and how to associate codes with non-probabilistic models. When using entropified model classes, the Shannon-Fano code can be justified in terms of minimizing the worst-case expected code length for probabilistic and non-probabilistic model classes alike. This makes ‘entropification’ a quite general means of turning model classes into codes. As such, it is in line with the general MDL philosophy, in which all models are viewed as probabilistic, or more properly, as codes. Let us consider a quote by Rissanen ([93], page 20)

‘... we then see that the unification obtained by interpreting all models

as probabilistic has given us an immutable yardstick, the code length, which we never can reduce to zero by scaling or other devices. The same cannot be said about the usually suggested prediction error measures, which we easily can scale to any size, and which therefore will never be able to serve as a universal yardstick for model selection.'

We agree that it is desirable and probably even possible to compare all models and model classes for given data  $D$  in terms of the code length they assign to  $D$ . But also we think that Rissanen's view leaves open two questions: first, how to base predictions and decisions on a 'probabilistic' model - since the main interpretation of the model is a code rather than a probability distribution according to which data are distributed, it is not *a priori* clear how this should be done. The second question is how to change model classes that are normally viewed as being 'non-probabilistic' into associated probabilistic model classes in a principled way. As we see it, the concept of 'reliable estimation' is a step towards answering the first question, while 'entropification' is a step towards answering the second question.





## Bibliography

- [1] H. Akaike, *Information theory as an extension of the maximum likelihood principle*, In B. N. Petrov and F. Csaki (Eds.), Second International Symposium on Information Theory, Budapest, pp. 267-281, 1973.
- [2] T.V. Allen, O. Madani, and R. Greiner, *Comparing model selection criteria for belief networks* Under submission, 2003.
- [3] L.J. Bain and M. Engelhardt, *Introduction to probability and mathematical statistics*, PWS-Kent, Boston, 1989.
- [4] V. Balasubramanian, *Statistical inference, Occam's Razor, and statistical mechanics on the space of probability distributions*, Neural Computation 9, pp. 349-368, 1997.
- [5] V. Balasubramanian, *MDL, Bayesian inference and the geometry of the space of probability distributions*, In P. D. Grünwald, I. J. Myung, and M. A. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications. MIT Press, 2004.
- [6] A.R. Barron, *Logically smooth density estimation*, Ph. D. thesis, Department of EE, Stanford University, Stanford, Ca, 1985.
- [7] A.R. Barron, *Complexity regularization with application to artificial neural networks*, In G. Roussas, editor, Nonparametric Financial Estimation and Related Topics, pp. 561-576, Dordrecht, Kluwer Academic Publishers, 1990.
- [8] A.R. Barron and T.M. Cover, *Minimum complexity density estimation*, IEEE Transactions on Information Theory 37, pp. 1034-1054, 1991.
- [9] A.R. Barron, J. Rissanen, and B. Yu, *The Minimum Description Length Principle in coding and modeling*, IEEE Transactions on Information Theory 44, pp. 2743-2760, 1998.
- [10] J. Bernardo, and A. Smith, *Bayesian theory*, John Wiley, 1994.
- [11] D. Bohm, *A suggested interpretation of the quantum theory in terms of 'hidden' variables I*, Physical Review 85, pp. 166-179, 1952.
- [12] D. Bohm, *A suggested interpretation of the quantum theory in terms of 'hidden' variables II*, Physical Review 85, pp. 180-193, 1952.
- [13] H. Bozdogan, *Personal communication*.
- [14] K.P. Burnham, and D. R. Anderson, *Model selection and multimodel inference*, New York: Springer-Verlag, 2002.
- [15] G. Casella, and R. Berger, *Statistical inference*, Wadsworth, 1990.
- [16] G.J. Chaitin, *Algorithmic information theory*, Cambridge University Press, 1991.
- [17] B.S. Clarke and A.R. Barron, *Jeffreys' prior is asymptotically least favorable under entropy risk*, Journal of Statistical Planning and Inference 41, pp. 37-60, 1996.
- [18] B.S. Clarke, *Comparing Bayes and non-Bayes model averaging when model approximation error cannot be ignored*, Under submission, 2002.
- [19] J.W. Comley, and D.L. Dowe, *Minimum Message Length and generalised Bayesian nets with asymmetric languages*, In P. D. Grünwald, I. J. Myung, and

- M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*, MIT Press, 2004.
- [20] T. Cover, and J. Thomas, *Elements of information theory*, New York John Wiley, 1991.
  - [21] I. Csiszár, and P. Shields, *The consistency of the BIC Markov order estimator*, The Annals of Statistics 28, pp. 1601-1619, 2000.
  - [22] A. Dawid, *Present position and potential developments: Some personal views, statistical theory, the prequential approach*, Journal of the Royal Statistical Society, Series A 147, pp. 278-292, 1984.
  - [23] A. Dawid, *Prequential analysis, stochastic complexity and Bayesian inference*, In J. Bernardo, J. Berger, A. Dawid, and A. Smith (Eds.), *Bayesian Statistics, Volume 4*, Oxford University Press, Proceedings of the Fourth Valencia Meeting, pp. 109-125, 1992.
  - [24] A. Dawid, *Prequential analysis*, In S. Kotz, C. Read, and D. Banks (Eds.), *Encyclopedia of Statistical Sciences, Volume 1 (Update)*, Wiley- Interscience, pp. 464-470, 1997.
  - [25] A. Dawid, and V.G. Vovk, *Prequential probability: Principles and properties*, Bernoulli 5, pp. 125-162, 1999.
  - [26] B. De Finetti, *Theory of Probability. A critical introductory treatment*, London: John Wiley & Sons, 1974.
  - [27] P. Domingos, *The role of Occam's razor in knowledge discovery*, Data Mining and Knowledge Discovery 3, pp. 409-425, 1999.
  - [28] D. Draper, *Assessment and propagation of model uncertainty*, Journal of the Royal Statistical Society, 1994.
  - [29] M. Feder, *Maximum entropy as a special case of the minimum description length criterion*, IEEE Transactions on Information Theory 32, pp. 847-849, 1986.
  - [30] W. Feller, *An introduction to probability theory and its applications*, volume 1, Wiley, 1968.
  - [31] R.A. Fisher, *On the mathematical foundations of theoretical statistics*, Philosophical Transactions of the Royal Society of London, Ser. A 222, pp. 309-368, 1922.
  - [32] D. Foster, and R. Stine, *Local asymptotic coding and the minimum description length*, IEEE Transactions on Information Theory 45, pp. 1289-1293, 1999.
  - [33] D. Foster, and R. Stine *The competitive complexity ratio*, In Proceedings of the 2001 Conference on Information Sciences and Systems. WP8 1-6, 2001.
  - [34] D. Foster, and R. Stine, *The contribution of parameters to stochastic complexity*, In P. D. Grünwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.
  - [35] D.A. Freedman, *A note on screening regression equations*, The American Statistician 37, pp. 152-155, 1983.
  - [36] P. Gács, *On the symmetry of algorithmic information*, Soviet Math. Dokl. 15, pp. 1477-1480, 1974.
  - [37] P. Gács, J. Tromp, and P. Vitányi, *Algorithmic statistics*, IEEE Trans. Inform. Th. 47, pp. 2443-2463, 2001.
  - [38] P.H. Garthwaite and J.M. Dickey, *Elicitation of prior distributions for variable selection problems in regression*, Annals of Statistics 20, pp. 1697-1719, 1992.

- [39] S. Geisser, *Discussion on sampling and Bayes' inference in scientific modelling and robustness*, Journal of the Royal Statistical Society A 143, pp. 416-417, 1980.
- [40] S. Geman, E. Bienenstock and R. Doversat, *Neural networks and the bias/variance dilemma*, Neural Computation 4, pp. 1-58, 1992.
- [41] E.I. George and R.E. McCulloch, *Variable selection via Gibbs sampling*, Journal of American Statistical Society 88, pp. 881-890, 1993.
- [42] I.J. Good, *Rational decisions*, Journal of the Royal Statistical Society B 14, pp. 107-114, 1952.
- [43] P.D. Grünwald, *A minimum description length approach to grammar inference*, In G. S. S. Wermter, E. Riloff (Ed.), *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, Number 1040 in Springer Lecture Notes in Artificial Intelligence, pp. 203-216, 1996.
- [44] P.D. Grünwald, *The Minimum Description Length Principle and reasoning under uncertainty*, PhD. thesis, University of Amsterdam, The Netherlands, 1998.
- [45] P.D. Grünwald, *Viewing all models as 'probabilistic'*, In Proceedings of the Twelfth Annual Workshop on Computational Learning Theory (COLT' 99), pp. 171-182, 1999.
- [46] P.D. Grünwald, *Maximum entropy and the glasses you are looking through*, In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI 2000), Morgan Kaufmann Publishers, pp. 238-246, 2000.
- [47] P.D. Grünwald, Manuscript CWI, 2003.
- [48] P.D. Grünwald, and A.P. Dawid, *Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory*, Annals of Statistics 32, 2004.
- [49] P.D. Grünwald, and J. Langford, *Suboptimal behaviour of Bayes and MDL in classification under misspecification*, In Proceedings of the Seventeenth Annual Conference on Computational Learning Theory, 2004.
- [50] P.D. Grünwald, I.J. Myung, and M.A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*, MIT Press, 2004.
- [51] M. Hansen, and B. Yu, *Wavelet thresholding via MDL for natural images*, IEEE Transactions on Information Theory 46, pp. 1778-1788, 2000.
- [52] M. Hansen, and B. Yu, *Model selection and the principle of minimum description length*. Journal of the American Statistical Association 96, pp. 746-774, 2001.
- [53] A.J. Hanson, and P.C.W. Fu, *Applications of MDL to selected families of models*, In P. D. Grünwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*, MIT Press, 2004.
- [54] D. Haussler, M. Kearns, H.S. Seung and N.Z. Tishby, *Rigorous learning curve bounds from statistical mechanics*, Machine Learning 25, pp. 195-236, 1996.
- [55] U. Hjorth, *Model selection and forward validation*, Scandinavian Journal of Statistics 9, pp. 95-105, 1982.
- [56] D.R. Hofstadter, *Gödel, Escher, Bach: An eternal golden braid*, Basic Books, 20th Anniv. Edition, 1999.
- [57] E. Jaynes, *Probability Theory: the logic of science*, Cambridge University Press. Edited by G. Larry Bretthorst, 2003.
- [58] H. Jeffreys, *An invariant form for the prior probability in estimation problems*, Proceedings of the Royal Statistical Society (London) Series A 186, pp. 453-461, 1946.

- [59] H. Jeffreys, *Theory in probability*, 3rd edition, Oxford University Press, 1961.
- [60] B.H. Juang, and L.R. Rabiner, *Hidden Markov models for speech recognition*, *Technometrics* 33, 251-272, 1991.
- [61] R.E. Kass, and A.E. Raftery *Bayes factors*, *Journal of the American Statistical Association* 90, pp 773-795, 1995.
- [62] R.E. Kass, and L. Wasserman, *The selection of prior distributions by formal rules*, *Journal of the American Statistical Association* 91, pp. 1343-1370, 1996.
- [63] R.E. Kass and P.W. Voss, *Geometrical foundations of asymptotic inference*, Wiley Interscience, 1997.
- [64] M. Kearns, Y. Mansour, A. Ng, and D. Ron *An experimental and theoretical comparison of model selection methods*, *Machine Learning* 27, pp. 7-50, 1997.
- [65] A.N. Kolmogorov, *Three approaches to the quantitative definition of information*, *Problems Inform. Transmission* 1(1), pp. 1-7, 1965.
- [66] A.N. Kolmogorov, *Talk at the Information Theory Symposium in Tallinn, Estonia*, according to P. Gács and T. Cover who attended it, 1974.
- [67] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri, *On supervised selection of Bayesian networks*, In K. Laskey and H. Prade (Eds.), *Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence*, 1999.
- [68] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri, *An MDL framework for data clustering*, In P. D. Grünwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.
- [69] A.D. Lanterman, *Hypothesis testing for Poisson versus geometric distributions using stochastic complexity*, In P. D. Grünwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.
- [70] P. Lee, *Bayesian Statistics - an introduction*, Arnold & Oxford University Press, 1997.
- [71] M. Li and P.M.B. Vitányi, *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, New York, 1997.
- [72] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi *The similarity metric*, In *Proc. 14th ACM-SIAM Symp. Discrete Algorithms*, 2003.
- [73] F. Liang, and A. Barron, *Exact minimax predictive density estimation and MDL*, In P. D. Grünwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*, MIT Press, 2004.
- [74] F. Liang, and A. Barron, *Exact minimax strategies for predictive density estimation*, To appear in *IEEE Transactions on Information Theory*, 2004.
- [75] D. Madigan and A.E. Raftery, *Model selection and accounting for uncertainty in graphical models using Occam's window*, *Journal of the american statistical society*, 1993.
- [76] D. Madigan and J. York, *Bayesian graphical models for discrete data*, Technical report 259, department of statistics, University of Washington, 1993.
- [77] N. Merhav and M. Feder, *Universal prediction*, *IEEE Trans. Inform. Theory* 44, pp. 2124-2147, 1998.
- [78] A.J. Miller, *Subset selection in regression*, New York: Chapman-Hall, 1990.
- [79] D.S. Modha, and E. Masry, *Prequential and cross-validated regression estimation*, *Machine Learning* 33, pp. 5-39, 1998.

- [80] I.J. Myung, V. Balasubramanian, and M.A. Pitt, *Counting probability distributions: Differential geometry and model selection*, Proceedings of the National Academy of Sciences USA 97, pp. 11170-11175, 2000.
- [81] D. Navarro, *Misbehaviour of the Fisher information approximation to Minimum Description Length*, Under submission, 2003.
- [82] E. Pednault, *Personal communication*, 2003.
- [83] J. Quinlan and R. Rivest, *Inferring decision trees using the minimum description length principle*, Information and Computation 80, pp. 227-248, 1989.
- [84] A.E. Raftery, *Approximate Bayes factors and accounting for model uncertainty in generalized linear models*, Technical Report 255, Department of Statistics, University of Washington, 1993.
- [85] H. Raiffa and R. Schlaifer, *Applied statistical decision theory*, Cambridge, MA, MIT Press.
- [86] L.E. Reichl, *A modern course in statistical physics*, Edward Arnold, London, 1991.
- [87] B. Ripley, *Pattern recognition and neural networks*, Cambridge University Press, 1996.
- [88] J. Rissanen, *Modeling by the shortest data description*, Automatica 14, pp. 465-471, 1978.
- [89] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, The Annals of Statistics 11, pp. 416-431, 1983.
- [90] J. Rissanen, *Universal coding, information, prediction and estimation*, IEEE Transactions on Information Theory 30, pp. 629-636, 1984.
- [91] J. Rissanen, *Stochastic complexity and modeling*, The Annals of Statistics 14, pp. 1080-1100, 1986.
- [92] J. Rissanen, *Stochastic complexity (with discussions)*, Journal of Royal Statistical Society 49, pp. 223-239, pp. 252-265, 1987.
- [93] J. Rissanen, *Stochastic complexity in statistical inquiry*, World Scientific Publishing Company, 1989.
- [94] J. Rissanen, T. Speed, and B. Yu *Density estimation by stochastic complexity*, IEEE Transactions on Information Theory 38, pp. 315-323, 1992.
- [95] J. Rissanen, *Fisher information and stochastic complexity*, IEEE Transactions on Information Theory 42, pp. 40-47, 1996.
- [96] J. Rissanen, *MDL denoising*, IEEE Transactions on Information Theory 46, pp. 2537-2543, 2000.
- [97] J. Rissanen, *Strong optimality of the normalized ML models as universal codes and information in data*, IEEE Transactions on Information Theory 47, pp. 1712-1717, 2001.
- [98] J. Rissanen, *Kolmogorov's structure function for probability models*, Proc. IEEE Information Theory Workshop. pp. 98-99, IEEE Press, 2002.
- [99] J. Rissanen, and I. Tabus *Kolmogorov's structure function in MDL theory and lossy data compression*, In P. D. Grunwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*, MIT Press, 2004.
- [100] G. Schwarz, *Estimating the dimension of a model*, The Annals of Statistics 6, pp. 461-464, 1978.
- [101] G. Shafer, and V. Vovk, *Probability and finance - It's only a game!* Wiley, 2001.

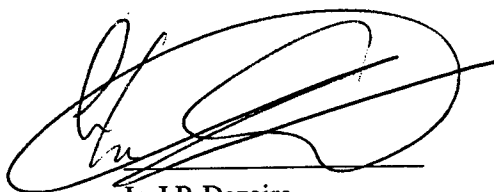
- [102] C.E. Shannon, *The mathematical theory of communication*, Bell System Tech. J. 27, pp. 379-423, 623-656, 1948.
- [103] A.Kh. Shen, *The concept of Kolmogorov  $(\alpha, \beta)$ -stochasticity and its properties*, Soviet Math. Dokl. 28 pp. 295-299, 1983.
- [104] Y.M. Shtarkov, *Universal sequential coding of single messages*, (translated from) Problems of Information Transmission 23, pp. 3-17, 1987.
- [105] A.F.M. Smith and G.O. Roberts, *Bayesian computation via Gibbs sampler and related Markov chain Monte Carlo methods*, Journal of the royal statistical society B 55, pp. 3-24, 1993.
- [106] R.J. Solomonoff, *A formal theory of inductive inference, part 1 and part 2*, Inform. Contr. 7, pp. 1-22, pp. 224-254, 1964.
- [107] R.J. Solomonoff, *Complexity-based induction systems: comparisons and convergence theorems*, IEEE Transactions on Information Theory 24, pp. 422-432, 1978.
- [108] T. Speed, and B. Yu, *Model selection and prediction: normal regression*, Ann. Inst. Statist. Math. 45, pp. 35-54, 1993.
- [109] J. Takeuchi, and A. Barron *Asymptotically minimax regret for exponential families*, In Proceedings SITA '97, pp. 665-668, 1997.
- [110] J. Takeuchi, and A. Barron, *Asymptotically minimax regret by Bayes mixtures*, In Proceedings of the 1998 International Symposium on Information Theory (ISIT 98), 1998.
- [111] J. Takeuchi, *On minimax regret with respect to families of stationary stochastic processes (in Japanese)*, In Proceedings IBIS 2000, pp. 63-68, 2000.
- [112] N.Z. Tishby, *Statistical physics models of supervised learning*, In D. Wolpert, editor, *The Mathematics of Generalization*, volume XX of SFO Studies in the Sciences of Complexity, pp. 215-242, Addison Wesley, 1995.
- [113] P. Townsend, *The mind-body equation revisited*, In C.-Y. Cheng (Ed.), *Psychological Problems in Philosophy*, pp. 200-218, 1975.
- [114] R. van Rooy, *Quality and quantity of information exchange*, Journal of Logic, Language and Information, 2003.
- [115] V. Vapnik, *Statistical learning theory*, John Wiley, 1998.
- [116] N.K. Vereshchagin and P.M.B. Vitányi, *Kolmogorov's structure functions and an application to the foundations of model selection*, Proc. 47th IEEE Symp. Found. Comput. Sci., pp. 751-760, 2002.
- [117] M. Viswanathan, C. Wallace, D. Dowe, and K. Korb, *Finding cutpoints in noisy binary sequences - a revised empirical evaluation*, In Proc. 12th Australian Joint Conf. on Artif. Intelligence, Volume 1747 of Lecture Notes in Artificial Intelligence (LNAI), Sidney, Australia, pp. 405-416, 1999.
- [118] P.M.B. Vitányi and M. Li, *Minimum description length induction, Bayesianism, and Kolmogorov Complexity*, IEEE Trans. Inform. Theory IT-46, pp. 446-464, 2000.
- [119] P.M.B. Vitányi, *Meaningful information*, In: Proc. 13th International Symposium on Algorithms and Computation (ISAAC), Vol. 2518 of Lecture Notes in Computer Science. Berlin, pp. 588-599, Springer Verlag, 2002.
- [120] P.M.B. Vitányi, *Algorithmic statistics and Kolmogorov's structure function*, In P. D. Grunwald, I. J. Myung, and M. A. Pitt (Eds.), *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2004.

- [121] C.S. Wallace, and D. Boulton, *An information measure for classification*, Computing Journal 11, pp. 185-195.
- [122] C.S. Wallace, and D. Boulton, *An invariant Bayes method for point estimation*, Classification Society Bulletin 3, pp. 11-34, 1975.
- [123] C.S. Wallace and P.R. Freeman, *Estimation and inference by compact coding*, Journal of the Royal Statistical Society B 49, pp. 240-251, discussion pp. 252-265, 1987.
- [124] G. Webb, *Further experimental evidence against the utility of Occam's razor*, Journal of Artificial Intelligence Research 4, pp. 397-417, 1996.
- [125] K. Yamanishi, *A decision theoretic extension of stochastic complexity and its applications to learning*, IEEE Transactions on Information Theory 44, pp. 1424-1439, 1998.
- [126] A. Zellner, *On assessing prior distributions and Bayesian regression analysis with g-prior distributions*, in Bayesian Inference and Decision Techniques-essays in Honor of Bruno de Finetti, North Holland, Amsterdam, pp. 233-243, 1986.
- [127] T. Zhang, *On the convergence of MDL density estimation*, In Y. Singer and J. Shawe-Taylor (Eds.), Proceedings of the Seventeenth Annual Conference on Computational Learning Theory (COLT' 04), Lecture Notes in Computer Science. Springer-Verlag, 2004.



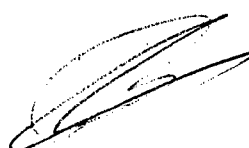


## Signature

A large, stylized handwritten signature in black ink, featuring a prominent loop and a long horizontal stroke.

Ir. J.P. Dezaire

Groepsleider

A handwritten signature in black ink, consisting of several fluid, connected strokes.

Ir. C.V. van Wijk  
Dr. Habil. H.W.L. Naus  
Auteur

A handwritten signature in black ink, featuring a series of overlapping loops and a long horizontal stroke.



# REPORT DOCUMENTATION PAGE

1. DEFENCE REPORT NO (MOD-NL) TD-04-0451	2. RECIPIENT'S ACCESSION NO	3. PERFORMING ORGANISATION REPORT NO TNO-DV1 2004 A234
4. PROJECT/TASK/WORK UNIT NO	5. CONTRACT NO	6. REPORT DATE April 2005
7. NUMBER OF PAGES 148	8. NUMBER OF REFERENCES 126	9. TYPE OF REPORT AND DATES COVERED Final
10. TITLE AND SUBTITLE Model selection and accounting for uncertainty		
11. AUTHOR(S) Ir. C.V. van Wijk and Dr. habil. H.W.L. Naus		
12. PERFORMING ORGANISATION NAME(S) AND ADDRESS(ES) TNO Defence, Security and safety, PO Box 96864, 2509 JG The Hague, The Netherlands Oude Waalsdorperweg 63, The Hague, The Netherlands		
13. SPONSORING AGENCY NAME(S) AND ADDRESS(ES)		
14. SUPPLEMENTARY NOTES The classification designation Ongerubriceerd is equivalent to Unclassified, Stg. Confidentieel is equivalent to Confidential and Stg. Geheim is equivalent to Secret.		
15. ABSTRACT (MAXIMUM 200 WORDS (1044 BYTE)) Statistical modeling is about finding general laws from observed data, which amounts to extracting information from the data. We are to admit no more causes of natural things (as we are told by Newton) than such as are both true and sufficient to explain their appearances. This central theme is basic to the pursuit of science, and goes back to the principle known as Occam's razor: 'if presented with a choice between indifferent alternatives, then one ought to select the simplest one.' Reliable inferences allow one to make good predictions and decisions regarding the data under a much wider variety of assumptions than unreliable inferences do. It will allow us to establish in what way we can and in what way we cannot use overly simple models. In general, we will be interested in what can be reliably predicted - and what not - from a model that is only partially correct. We describe a new procedure called entropification. With an entropified model, if given enough data, we can find the model with the smallest expected prediction error. This model will provide a correct estimate of the average prediction error that it will achieve; hence the model gives a good impression of 'how good it really is.'		
16. DESCRIPTORS Military vehicles, Magnetic signatures		IDENTIFIERS Modeling, Reliable inference, Entropification
17a. SECURITY CLASSIFICATION (OF REPORT) UNCLASSIFIED	17b. SECURITY CLASSIFICATION (OF PAGE) UNCLASSIFIED	17c. SECURITY CLASSIFICATION (OF ABSTRACT) UNCLASSIFIED
18. DISTRIBUTION AVAILABILITY STATEMENT Unlimited		17d. SECURITY CLASSIFICATION (OF TITLES) UNCLASSIFIED



# Distributielijst

## Distributie rapport

1. SC-WOO
2. OTC Genie/Kenniscentrum, t.a.v. Maj L. Lagerwerf  
(programmaleider "mijnen" V010)
3. OTC Genie/Kenniscentrum, t.a.v. Lkol. Th.G.M. van Wijk
4. OTC Genie/Kenniscentrum, t.a.v. Kap. C.M. Netten
5. NATCO/LBBKL/ABWM, t.a.v. Lkol ing. W. de Jong
6. NATCO/LBBKL/Munitiebedrijf, t.a.v.  
Mw. Ing. N.L.P. de Bruyn Prince - van Kempen
7. DMKM/MARTECH, t.a.v. KLTZE ir. W.W. Schalkoort
8. LAS/BP/BO, t.a.v. Lkol J.W.T.M. Mekers (prog. beg. "Pantservoertuigen" V022)
9. LAS/OBS, t.a.v. Lkol. G. van Grunsven
10. TNO-DO, t.a.v. Ir. Z.J.G. Groh (programmaleider "Pantservoertuigen" V022)
11. TNO Defensie en Veiligheid, Directeur Kennis, daarna reserve
12. TNO Defensie en Veiligheid, Directeur Operaties, daarna reserve
13. Bibliotheek KMA
14. Bibliotheek KMA
15. Bibliotheek KMA
16. Archief TNO Defensie en Veiligheid, in bruikleen aan Dr.ir. A.L.D. Beckers
17. Archief TNO Defensie en Veiligheid, in bruikleen aan Ir. J.P. Dezaire
18. Archief TNO Defensie en Veiligheid, in bruikleen aan Dr. Habil. H.W.L. Naus
19. Archief TNO Defensie en Veiligheid, in bruikleen aan Ir. J.B. Oostinga
20. Archief TNO Defensie en Veiligheid, in bruikleen aan Dr. A.J. Schoolderman
21. Archief TNO Defensie en Veiligheid, in bruikleen aan Dr. Ir. S.H.J.A. Vossen
22. Archief TNO Defensie en Veiligheid, in bruikleen aan Ing. F.J. de Wolf
23. Documentatie TNO Defensie en Veiligheid
24. Reserve

## Distributie managementuittreksel & distributielijst

- 1x TNO Defensie en Veiligheid, Algemeen directeur, TNO Defensie en Veiligheid,  
Directeur Markt, daarna reserve, MIVD / AAR / HBMT
- 4x SC-WOO, Hoofdcluster Kennistransfer, Kol. A.P. Coppens